DIVISION OF
SOLID MECHANICS
STRUCTURES
AND
MECHANICAL DESIGN

REPORT NO. 24

AN ALGORITHM FOR THE MINIMUM WEIGHT DESIGN OF THE GENERAL TRUSS

BY

MILTON J. SCHRADER

JUNE 1968

SCHOOL OF ENGINEERING • CASE WESTERN RESERVE UNIVERSITY

Program supported in part by the National Aeronautics and Space Administration Under Research Grant No. NsG 110-61

AN ALGORITHM FOR THE MINIMUM WEIGHT DESIGN OF THE GENERAL TRUSS

Report No. 24

by

Milton J. Schrader

June 1968

Approved by:

Richard L. Fox
Assistant Professor
of Engineering

SOLID MECHANICS,
STRUCTURES AND
MECHANICAL DESIGN

FOREWORD

This report is a reproduction of a thesis submitted to Case Western Reserve University in partial fulfillment of the requirements for the Degree of Masters of Science.

This work is part of the continuing effort in the application of structural synthesis to aerospace vehicle structures supported in part by the National Aeronautics and Space Administration under Grant No. NsG 110-61.

An Algorithm for the Minimum Weight Design

of the General Truss

by

Milton James Schrader

ABSTRACT

An algorithm is presented for the minimum weight design of a three dimensional, linear truss. The truss has an arbitrary number of nodes, members, and imposed load conditions. The members are assumed to be tubular and are described by the mean diameter and wall thickness. The truss is subjected to constraints on the nodal displacements, member stresses, and member sizes. More specifically, the stresses are limited by the yield strength of the material and the crippling and Euler buckling stress; the member sizes are limited by maximum values on the diameters and thicknesses, the upper and lower ratio of the diameter to thickness, and the fact that the diameters and thicknesses must remain positive. Also, the ability to link the various design variables is included.

The method, based on the integrated approach to synthesis and analysis, places an upper constraint on the design weight and transforms the original problem into a residual minimization subject to the original constraints plus the weight constraint. This new problem, cast into the form of an unconstrained

minimization problem by using the Fiacco-McCormick technique, is solved for a series of decreasing values of the weight constraint until a near optimum design is reached.

Results for several examples were obtained. These results were compared with the results from other methods whenever possible to demonstrate the effectiveness of the algorithm.

ACKNOWLEDGMENTS

The author wishes to express his sincerest graditude to:

Professor Richard L. Fox, whose advice, suggestions, and encouragement was an essential element in the completion of this thesis,

Miss Diane Dryfuse for the excellent typing of this manuscript anc also, for her patience,

The personnel of the Case Computing Center and Mr. Charles Kaine for their help in obtaining the numerical results for this thesis,

The National Aeronautics and Space Administration for their support under grant No. NsG 110-61.

LIST OF SYMBOLS

Symbol	Description
A_i	area of the i^{th} member
C_N	conditioning number
C_{z1}	a nondimensionalizing factor
D	an n by n nonsingular matrix used for scaling transformations
D_i	a preassigned constant for the i^{th} member
DD_j	a set of integers
d_i	diameter of the i^{th} member
d_{max}	maximum allowable diameter
E	Young's modulus
g_j	the j^{th} constraint
K_d, K_{uz}	nondimensionalizing factors
K_{ij}	element of the master stiffness matrix
$\tilde{K}_{ivb}$	element of the local stiffness matrix for the i^{th} member

Symbol	Description
KT	total number of loading conditions
k_2	crippling constant
L_i	length of the i^{th} member
M	total number of truss members
N	total number of degrees of freedom per loading
N_{dv}, N_{tv}	number of diameter and thickness variables
NN	total number of nodes of the truss
nc	total number of constraints
P	load vector(s)
P_{jk}	load on the j^{th} degree of freedom for the k^{th} load condition
PT_i	a set of integers
R	the sum of the squares of the residual; also referred to as the residual
r_{jk}	residual of the j^{th} degree of freedom for the k^{th} load condition; sometimes called the residuals
r_ℓ, r_u	lower and upper limit of the diameter to thickness ratios

Symbols	Description
SR_{ij}	a set of integers
T_i	a preassigned constant for the i^{th} member
TT_i	a set of integers
t_i	thickness of the i^{th} member
t_{max}	maximum allowable thickness
U	displacement vector(s)
UU_i	a set of integers
u_{jk}	displacement of the j^{th} degree of freedom for the k^{th} load condition
u_{low}, u_{up}	lower and upper limits on the displacements
W	weight of the truss
W_o	value of the weight constraint; also called the goal weight
$\overline{X}$	vector of synthesis variables
$\overline{x}^*$	the minimum of a function
α_j	the j^{th} diameter parameter
β_j	the j^{th} thickness parameter

Symbol	Description
γ_{ic}	the direction cosine of the i^{th} member with respect to the c axis
ε_M	tolerance on the residuals
$\eta\binom{i}{j}$	a function defined to have a value of 1 or 0; it is 1 only if i equals j
$\lambda_{max}, \lambda_{min}$	maximum and minimum eigenvalues
ν	the multiplier for the Fiacco McCormick formulation
ρ	unit weight of the material
σ_b	Euler buckling stress
σ_c	crippling stress
σ_{ik}	stress in the i^{th} member for the k^{th} load condition
σ_{yc}, σ_{yt}	the compressive and tensile yield stresses
ϕ	the integrated analysis synthesis function

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

One aspect of structural design which has received considerable interest and study in recent years is the approach known as the structural synthesis process. This activity might be described as a rational process directed toward producing a design which is better than any other allowable design in achieving some quality and which satisfies a set of requirements or specifications. Typical of such qualities are the weight and cost of the structure being designed.

In the field of structural synthesis, the structural system which probably has been studied the most is the truss. A truss will be defined as a structure composed of bars which are hinged together at their ends to form a stable, rigid framework. The usual assumptions will be made considering the system:[1]

1. The nodes where the members are joined together are frictionless pin joints.
2. Loads and reactions are applied at the nodes.
3. The members are subjected only to axial forces (e.g. no bending moments are considered to occur).

---

[1]Charles Head Norris and John Benson Wilbur, <u>Elementary Structural Analysis</u>, Second Edition(New York: McGraw-Hill Book Company, 1960), p. 115.

Furthermore, consideration will only be extended to the linear truss. This requires that nodal displacements are small enough that the change in geometry has no significant effect on the analysis.

Since it would appear that the truss is a popular system with which to develop and study methods of structural synthesis, some of the reasons for its popularity should be listed. Among these the following might be included:[1]

1. It is possible to encounter a large number of degrees of freedom which is a characteristic of many interesting structural systems.
2. Methods of analysis are well developed.
3. There is a large choice of possible behavior requirements of varying complexity.

The following is a description of research into the further development of structural synthesis of the truss. In particular, it is a development within the framework of the integrated approach to engineering design as formulated by R. L. Fox and L. A. Schmit (see Reference 1).

The specific problem is to minimize the weight of a general space truss which is composed of M tubular members, joined at

---

[1] Richard L. Fox, "An Integrated Approach to Engineering Synthesis and Analysis", (Ph.D. Thesis, Engineering Division, Case Institute of Technology, 1965), p. 22.

NN nodes with N total nodal degrees of freedom or displacements for each load condition, and subjected to KT load conditions or configurations. The i^{th} truss member will be described by two dimensions, namely the mean diameter d_i and wall thickness t_i. The location of the j^{th} node is given by a set of coordinates (x_j, y_j, z_j).

The analysis of the truss, that is, the prediction of the displacements and the stresses within the truss due to the sets of applied loads, will be accomplished by the displacement method. In general, this method can be written in the matrix form

$$[k_{ij}]U = P \tag{1.1}$$

for each load condition. The matrix $[k_{ij}]$ is called the stiffness matrix and is computed from the geometry of the truss, the Young's modulus and cross sectional design of the members. The element k_{ij} is the force at the i^{th} degree of freedom due to a unit displacement at the j^{th} degree of freedom with all other possible displacements held at zero.

As an illustration of the formulation, the displacement equations for the three bar truss subjected to the two loading conditions shown in Figure 1 are

$$\left(\frac{A_1 E}{L_1}\right) U_{1x} + \left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right) U_{1y} = P_{1x}$$

$$\left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right) U_{1x} + \left(\frac{A_1 E}{2L_1} + \frac{A_2 E}{L_2} + \frac{A_3 E}{2L_3}\right) U_{1y} = P_{1y}$$

$$\left(\frac{A_1E}{L_1}\right)U_{2x} + \left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right)U_{2y} = P_{2x}$$

$$\left(\frac{EA_1}{2L_1} - \frac{EA_3}{2L_3}\right)U_{2x} + \left(\frac{A_1E}{2L_1} + \frac{A_2E}{L_2} + \frac{A_3E}{2L_3}\right)U_{2y} = P_{2y} \qquad (1.2)$$

where

A_i is the area of member i

E is the Young's modulus

L_i is the length of member i

P_{kj} is the jth component of the kth load

Unfortunately, most sets of displacement equations are not as simple to solve as the one presented above.

It has been mentioned that the final design of the truss must satisfy certain requirements or specifications. First, maintenance of structural integrity is an important requirement. To maintain structural integrity may mean that the stresses within each member be bounded by certain upper and lower limits. The material's tensile and compressive yield strengths might provide two possible limits of this type. Other limits on the stress such as the crippling stress and Euler's buckling stress could be dependent on the member's cross sectional dimensions. Second, limits are often prescribed for the displacement of the nodes. Occasionally, the function of the truss will require that limits on the displacements be specified. Also, the consideration of a linear truss necessitates some restrictive values for the displacements. A third set of requirements may be derived from items which are not associated with the stresses and displacements of the truss. These are usually due to practical factors. For example, the ability of the production process may place a lower limit on the size to which the wall thickness can be made.

The set of conditions or constraints which will be considered here are the following:

1. All diameters d_i and thicknesses t_i will be less than the specified maximum values d_{max} and t_{max} respectively.
2. The diameters and thicknesses are strictly positive.
3. The ratios of member diameter to thickness d_i/t_i should remain between the given upper and lower limits r_u and r_ℓ .
4. The displacements are to be within the upper and lower limits u_{up} and u_{low} respectively.
5. The member stresses are less than the material tensile yield stress and larger algebraically than the greatest of the material compressive yield stress, the Euler buckling stress, and the crippling stress.

These conditions represent some of the most common constraints placed on the design of a structure of this type.

Another requirement which will be imposed is that some of the design variables may be preassigned to have a constant value throughout the synthesis procedure. This would allow the thickness or diameter or both to be given prescribed values from the start of the design.

Furthermore, an additional requirement will be that the design of some members is made to depend directly on the design of another member by a constant ratio. This feature is called design variable linking. Linking provides the ability to require that some members of the truss be made the same and is a design practice which is often used to reduce the time or cost of fabricating the structural members and erecting the structure.

For the case of the three bar truss shown in Figure 1, the following constraints and conditions will be imposed as an example of the above discussion:

1. The stress in the i^{th} member satisfy

$$\sigma_{yt} > \sigma_i > \max(\sigma_{yc},\sigma_b,\sigma_c) \tag{1.3}$$

where

σ_b is the Euler buckling stress given by

$$\sigma_b = - \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \tag{1.4}$$

σ_c is the local buckling or crippling stress given by

$$\sigma_c = \frac{-k_2 E_i t_i}{d_i} \tag{1.5}$$

If σ_{yt} and σ_{yc} are 20 ksi and -15 ksi respectively, then, the constraints can be written as

$$20000 - \sigma_i > 0 \qquad i = 1,2,3$$

$$\sigma_i + 15000 > 0 \qquad i = 1,2,3$$

$$\sigma_i + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} > 0 \qquad i = 1,2,3$$

$$\sigma_i + \frac{k_2 E_i t_i}{d_i} > 0 \qquad i = 1,2,3$$

2. The j^{th} degree of freedom under the k-th load is limited by

$$u_{up} > u_{jk} > u_{low} \tag{1.6}$$

If u_{up} and u_{low} are ± 1.0, the constraints would appear as

$$u_{jk} + 1.0 > 0 \qquad j = x,y \quad , \quad k = 1,2$$

$$1.0 - u_{jk} > 0 \qquad j = x,y \quad , \quad k = 1,2$$

3. All the thicknesses will have a value of $1/4\pi$ inches.

4. The diameters of members 1 and 3 be linked so that

$$d_1 = d_3$$

5. The diameters are to be no larger than 10 inches and must be positive or

$$10 - d_i \geq 0 \quad , \quad i = 1,2,3 \tag{1.7}$$

$$d_i \geq 0 \quad , \quad i = 1,2,3 \tag{1.8}$$

6. The diameters will be no smaller than the thickness or

$$\frac{d_i}{t_i} - 1.0 \geq 0 \tag{1.9}$$

Perhaps at this point it should be emphasized that comparison of designs which satisfy the constraints is based upon the weight of the structure. The selection of weight as a criteria for comparison is common. Often the cost of the structure is assumed to be greatly dependent on the weight of the material used. This assumption appears to be an adequate way of simplifying a cost optimization problem in some cases. For application in aircraft and missiles, it is considered important to reduce the weight of the structural framework for the craft to achieve higher performances. Thus, for two designs which satisfy the constraints, the better design will be the one with the lower weight. For the case of the three bar truss of Figure 1, the weight is computed by

$$W = 7.5\rho(2.828\, d_1 + d_2) \tag{1.10}$$

A number of methods for solving the minimum weight truss problem have been presented. Bibliographies listed in Reference 5 published in 1958, Reference 6 published in 1963, and Reference 7 published in 1965 give an extensive survey of the work in the general field of structural optimization which includes the truss problem. Many methods are not applicable for the general truss with the constraints considered in this paper. Some solutions fail to consider all of the constraints such as displacement limitations, others are valid for only statically determinate trusses, and a number handle only a single load condition.

For statically determinate trusses which are stress limited, the approach by F. R. Shanley, Reference 2, is quite useful. In this solution, the analysis of the truss is made and the member forces are determined. These forces are independent of the cross sectional areas of the members. The proportioning of each member is carried out for the critical force in the member so that the member does not fail due to violations of the stress constraints. If the stress constraints consider only the material strength of the material, the design simply needs to supply the member's cross sectional area. If the crippling and Euler buckling failure modes are also taken into account, then the member's cross sectional dimensions must be specified since the stress limits and dimensions are related. The basic problems associated with this solution are that it is effective only for statically determinate structures and it fails to account for constraints other than those of stress.

Another class of approaches called the method of alternate steps has received much study. In this approach, the initial design satisfies all the conditions imposed on the truss and has a weight which is greater than the optimum weight of the truss. The weight is then reduced along the gradient of the weight function. This is called the path of steepest descent and is orthogonal to surfaces of equal weight. This reduction is carried out until a constraint is reached to within some tolerance δ . Then a redesign, called a side step, is made on the constant weight surface in an effort to move away from the constraints. After the side step has been completed, another move along a path of steepest descent is taken until a constraint is reached once again. This cycle of steepest descent and side step is repeated until the minimum weight design is reached.

One method of alternate steps was presented by R. A. Gellatly, R. H. Gallagher, and W. A. Luberacki (see Reference 3). The size and direction of the side step is determined by the curvature of the constraints in the vicinity of the design. Another variation of the method, presented by L. A. Schmit and R. H. Mallett in Reference 4 is particularly suited to problems in which the weight surfaces are nonlinear. When a constraint is broken in the path of steepest descent and the last acceptable design is not too close to the constraints, then a steepest descent path is initiated from the last acceptable design.

It is worth noting that in moving along the path of steepest descent and in locating a suitable side step, numerous designs must be analyzed in order to check the displacement and stress constraints. While for small systems this presents little worry, it is very time consuming for the case of a system with many degrees of freedom.

Another method is that of the integrated approach to engineering synthesis and analysis presented by R. L. Fox and L. A. Schmit, References 1 and 8. Simultaneous evolution of a design and its analysis is the basic aim of the approach. For the truss problem, the formulation requires a goal or drawdown weight W_0 which is made an upper limit of a weight constraint written as

$$W_0 - W > 0 \tag{1.11}$$

A penalty function is then formed. It is a measure of how greatly the constraints are violated. A residual is also created to evaluate the degree to which the analysis is not satisfied. Finally, another function is constructed from the combination of the penalty function and the residual. By minimizing this new function, an analyzed design can be found if the goal weight is not lower than the optimum weight. As analyzed designs are found, the goal weight is reduced until it is impossible to find an analyzed design by the minimization. When this happens, the last design is accepted as an approximation to the optimum. Development of the work presented in this paper was made within the philosophy of the integrated synthesis approach.

CHAPTER II

MATHEMATICAL PROGRAMMING FORMULATION OF STRUCTURAL SYNTHESIS

For the purpose of this paper, structural synthesis will be defined as a rational design procedure of a structural system that, according to a definite objective, produces a design which efficiently fulfills a set of specified functions. Some of the characteristics of the methods of structural synthesis are: the absence of a priori assumptions of the failure modes for various load conditions; absence of before hand knowledge about the critical load condition for a given member; and the idea of designing the structure as an integral unit and not as a group of individual subsystems.

For the general truss as with many other structural forms, structural synthesis consists of three elements. These components are the analysis of the behavior of the design, a set of constraints, and finally, an objective or merit function which is the basis of comparison between different designs. When the elements are related in an algebraic manner, the synthesis may be considered as a mathematical programming problem which has an objective function and constraints on the behavior and design variables.

The prediction of the behavior of the structure has two main parts. They are the selection of the method of analysis and the specification of the loading conditions applied to the structure.

For the truss, several methods of analysis have been developed. The displacement, force, and force-displacement methods exemplify the variety of approaches available. Factors which have a bearing on the selection of the appropriate method include the numerical stability of the various approaches, the computer storage space required by the different methods, and the accuracy needed for the solution.

Once the selection of the method of analysis has been made, attention can be turned to the loading conditions which will be considered in the analysis. Specification of the loading conditions are commonly given in a deterministic manner. That is, the problem is usually stated so that a discrete set of loading conditions are applied to the structure to represent all of the potentially critical loadings which the structure will ever encounter. This is different from reality where loads may be changing continuously and where the magnitude of the loads have some statistical distribution.

Great care must be used in selecting the design constraints. Certainly some restrictions are obvious, however, many structural failures are due to those constraints which managed to escape the engineer's attention.

Generally constraints may be divided into two groups. The first group may be called the behavior constraints which are those restrictions that must be imposed to maintain satisfactory behavior of the structure under the applied loads. An example of such constraints is the requirement that the truss member stresses be within the stress limits provided by the material's strength. Of course, assuring that the behavior constraints are satisfied requires that an analysis of the structure be carried out.

The second group of restrictions may be called the side constraints. These constraints include requirements on the size or proportioning of the design variables due to reasons not explicitly related to the behavior requirements. That the diameters of the truss members be limited to some maximum value may serve as an example. The motivation for such a constraint might be the desire to guarantee a sufficiently large length to diameter ratio to insure some architectural effect. The constraint given in the example is called an inequality constraint and can be written as either

$$d_{max} > d_i \tag{2.1a}$$

or

$$d_{max} - d_i > 0 \tag{2.1b}$$

Linking of the design variables may also be considered a side constraint. This might appear as an equality constraint. That is, member i of a truss could be made to have the same diameter as member j. This would be written as

$$d_i = d_j \tag{2.2}$$

Such a requirement is probably due to some external reason such as the cost or ease of construction.

It should be noted that constraints may be either linear or nonlinear. An example of a linear constraint is the requirement that the diameter of a truss member be less than some maximum value. An example of a nonlinear constraint is that the Euler buckling stress be less than the stress in the member or

$$\sigma - \sigma_b > 0$$

where, for a tubular member,

$$\sigma_b = -\frac{\pi^2 E(d^2 + t^2)}{8L^2}$$

σ is the stress in the member.

Generally, it is harder to work with the nonlinear constraints and, unfortunately, nonlinear constraints seem to occur more often than linear constraints in real structural problems.

The objective function is the basis of distinction between different designs which satisfy the constraints. Common ones include the weight or cost of the structure. Other possible objective functions are aerodynamic performance or heat transfer rate.

The importance of the selection of the loading system and constraints should be emphasized. The omitting of a critical load condition or neglecting an important failure mode is the usual cause of structural failures. No matter how efficient the synthesis procedure is, if an active load condition is not recognized, or if a significant failure mode is overlooked, the final design is apt to have unsatisfactory behavior if not collapse.

Once the analysis, constraints, and objective function have been specified, the formulation of structural synthesis as a mathematical programming problem may be accomplished.

Basic to the formulation is a set of quantities which are allowed to vary throughout the synthesis. These quantities are called design variables. Those quantities which do not vary and are given values at the initiation of the process are called preassigned parameters. When each of the design variables are assigned an axis in an n-dimensional Cartesian space, the space will be known as a design variable space and a given set of values of the design variables are represented by a point in that space.

In the design variable space, all points at which there is incipient failure to satisfy a particular requirement form a constraint surface. If the design is slightly on one side of this surface, the design fails to satisfy that particular constraint, while if slightly on the other side of the surface, the design does satisfy that constraint. If the constraint surface is determined by behavioral constraints, then the surface is called a behavior constraint surface. Otherwise, it may be said to be a side constraint surface. In structural problems there is a behavior constraint surface for each load condition for a particular failure mode. If there are more than one requirement on the design, which is the most common case, then the collection of all constraint surfaces is called the composite constraint surface.

Designs which are within the composite constraint surface, that is, designs that fulfill all the requirements, are deemed acceptable designs. If even one constraint is not satisfied by a design, it is called an unacceptable design. A design which is on the constraint surface (or at least within a region ϵ from the surface) is called a bound point and the surface is considered to be an active constraint. If a point is not a bound point, then it is a free point.

With the above description, the general truss problem presented in Chapter I may be stated as follows:

Given: preassigned parameters, the loading system with one or more distinct load conditions, and behavior and side constraints;

Find: the set of design variables subject to the side and behavior constraints such that the weight W is given a minimum value.

As an example of some of the terms mentioned in the previous discussion, the design space of the three bar truss problem discussed in Chapter I will be described. In Figure 2, the coordinate axes are labeled for the member diameters d_1 and d_2. Note that the area of member i is given by

$$A_i = \pi t_i d_i \tag{2.3}$$

Since the presentation of the example in Chapter I required

$$t_i = 1/4\pi$$

then

$$A_i = d_i/4 \tag{2.4}$$

Curves 1, 2, and 3 correspond to the compressive and tensile stress constraints. In particular, Curve 1 is associated with

$$\sigma_{11}(d_1,d_2) = \sigma_{32}(d_1,d_2) = -15000 \tag{2.5a}$$

Curve 2 is associated with

$$\sigma_{12}(d_1,d_2) = \sigma_{31}(d_1,d_2) = 20000 \tag{2.5b}$$

and Curve 3 is associated with

$$\sigma_{21}(d_1,d_2) = \sigma_{22}(d_1,d_2) = 20000 \tag{2.5c}$$

where

σ_{jk} indicates the stress in the j^{th} member for the k^{th} load condition.

There is an additional constraint that the diameters of the members must be positive. This makes the d_1 and d_2 axes constraints also. The requirement that the diameters are positive is a side constraint; the demands placed on the stresses are behavior constraints. All of the constraints form the composite constraint surface. The various points A, B, C, D, and E of Figure 2 can be described in the following manner:

A - free, acceptable

B - free, unacceptable

C - bound, acceptable

D - bound, unacceptable

E - optimum design

Curve 4 represents a weight contour. Specifically, it is a line of designs which all weigh 150 pounds if ρ is equal to unity. Curve 5 is another weight contour for designs weighing 100 pounds. Indeed, any line parallel to Curves 4 or 5 will be a weight contour. The weight for this problem is an example of a linear objective function and can be expressed in the form

$$W = \sum_{i=1}^{3} k_i D_i \qquad (2.6)$$

where

D_i is the diameter of the member i

k_i is a constant.

The assertion that nonlinear constraints are more difficult to work with than linear constraints has been made. For the problem with linear constraints, one method of solution is to move from one intersection of constraints to another in such a way so as to reduce the objective function. Since, for linear objective functions, an optimum is at one of these intersections, the approach will converge to the solution. For problems with nonlinear constraints, movement from intersection to intersection is quite difficult. Furthermore, there is no guarantee that the solution is at an intersection.

Due to the difficulties with the nonlinear constraints, various approaches for optimization such as the method of alternate steps have been developed. Regrettably, the use of these approaches necessitates numerous analyses of the designs along the synthesis path. To avoid this, the integrated approach to synthesis and analysis was developed. The formulation of a method within the philosophy of the integrated approach is presented in the next chapter.

CHAPTER III

FORMULATION OF THE SOLUTION

The problem of optimizing the weight of a general truss has been presented in Chapter I with a description of the constraints and the method for analysis. To illustrate the different aspects of the proposed synthesis, an example of a three bar truss with various constraints was given. A more concise statement of the problem is

Given: preassigned parameters, the loading system with one or more distinct load conditions, and behavior and side constraints;

Find: the set of design variables subject to the side and behavior constraints such that the weight W is given a minimum value.

Some of the difficulties associated with the many approaches to the problem were mentioned in the previous chapters. For those approaches which are able to handle a variety of load conditions and constraints, the solutions require numerous analyses. This difficulty is emphasized whenever the system has a large number of degrees of freedom. In an effort to avoid this problem, the integrated approach to synthesis attempts to perform the design and analysis simultaneously. Thus, the synthesis variables include both the behavior and member design variables. The behavior variables consist of the displacements of the nodes, stresses within the members, or some other set of quantities which can fully describe the behavior of the structure under all the loading conditions. The member design variables are those parameters which specify the cross sectional dimensions of the truss members. Attention should be drawn to the fact that the number of behavior variables is dependent on the method of analysis.

A. Formulation of Method

Since the integrated approach attempts to evolve designs and their analyses concurrently, it would seem desirable that the new design have a weight which is less than the weight of the previously analyzed designs. One method to accomplish this is presented in the development of the integrated approach by R. L. Fox in Reference 1. The method is to select a goal weight W_0 which is made the upper limit of a constraint on the weight written as

$$g_W = W_0 - W \geq 0 \tag{3.1}$$

Once this is done, the weight is treated as any other constrained parameter and any analyzed design which satisfies all the constraints will be at most equal to the goal weight. If the goal weight is selected so that it is progressively reduced, then the weights of the designs will be successively smaller.

After the weight constraint has been imposed, the primary consideration is the development of a method to analyze a design and a procedure to keep within the constraints. These involve the construction of a measure of the degree to which the analysis is satisfied and selection of a standard mathematical programming technique.

The displacement method of analysis given by Equation (1.1) can be written as

$$f_i(\bar{x},\bar{D}) = 0 \quad , \quad i = 1,2,\ldots,n \tag{3.2}$$

where $\bar{x}$ is a unique behavior for the design variables $\bar{D}$. Since such a form does exist, then one method of finding the solution $\bar{x}_j$ for a given design $\bar{D}_j$ is

1. Define a function called the residual as

$$R(\bar{x}) = \sum_{i=1}^{n} [f_i(\bar{x},\bar{D})]^2 \tag{3.3}$$

2. Find $\bar{x}$ such that

$$R(\bar{x}) \rightarrow MIN = 0$$

The minimization of $R(\bar{x})$ can be accomplished by using one of the minimization schemes developed for an unconstrained function. Two possible schemes, the Fletcher-Powell and the Fletcher-Reeves methods, will be described in another chapter.

In the synthesis process, both the behavior variables $\bar{x}$ and the design variables $\bar{D}$ are allowed to vary at the same time. If

$$\bar{X} = \left(\frac{\bar{x}}{\bar{D}}\right)$$

then the problem of finding $\overline{X}$ such that $R(\overline{X})$ is reduced to zero has no unique solution. For the displacement method of analysis, the residual is defined as

$$R(\overline{X}) \quad \sum_{k=1}^{KT} \sum_{i=1}^{N} r_{ik}^2 \tag{3.4}$$

where

$$[r_{ik}] = [K_{ij}][u_{jk}] - [P_{ik}] \tag{3.5}$$

The element r_{ik} will be called the residual of the i^{th} degree of freedom for the k^{th} load condition or simply a residual. If there is possible confusion due to the common usage of the word "residual", the meaning will be made clear by writing the appropriate symbol for the quantity in question.

Another consideration is the criteria for which the equations of analysis will be considered satisfied. One possible criterion can be written as

$$R \leq \varepsilon_R$$

This requirement is not particularly adequate since it does not provide control over the individual residuals r_{ik}. That is, if all the residuals are zero except for one, say the pq^{th} element, then the analysis might be accepted even though the pq^{th} residual is quite large. To avoid this, a criterion which provides direct control on the magnitude of the individual residuals was adopted.

The condition is of the form

$$|r_{ik}| \leq r_M \qquad \begin{aligned} i &= 1,2,\ldots,N \\ k &= 1,2,\ldots,KT \end{aligned} \tag{3.6}$$

In the statement of the problem there is a set of behavior and side constraints. Besides these constraints, there is the additional constraint imposed on the weight by Equation (3.1). The convention that the inequality constraints be written so as to be positive when the restrictions are satisfied will be adapted. For example,

$$g_k = d_{max} - d_i \geq 0$$

is a constraint on the maximum size of the diameter of the i^{th} member.

At this point the problem may be restated as

Given: preassigned parameters, behavior and side constraints, the loading system with one or more distinct loading conditions, and the weight constraint;

Find: the set of behavior and design variables subject to the behavior, side, and weight constraints such that the residual R is given a minimum value.

If at the minimum of this problem the analysis cannot be satisfied, then the goal weight is less than the optimum weight.

The minimization problem presented above, which is merely a transformation of the original problem, can be cast into the form of a Fiacco-McCormick function. That is, construct the function

$$\phi(X,\nu) = R(X) + \nu\left[\sum_{i=1}^{nc} \frac{a_i}{g_i(X)} + \frac{a_w}{g_w(X)} \right] \tag{3.7}$$

where $(\nu > 0)$ and a_i and a_w are constants. Choose an initial multiplier $\nu > 0$ and an acceptable point X_0 which is strictly within the constraint set. If X should fall outside of the constraint set, then let the function have a value

$$\phi(X,\nu) \to + \infty$$

Intuitively, it would seem likely that the minimum would always fall within the constraint set. Thus, the problem is one of minimizing ϕ starting at X_0 using an unconstrained minimization scheme. Let the minimum of $\phi(X,\nu_i)$ be a point denoted by $X(\nu_i)$.

Once $X(\nu_1)$ is determined, the process is repeated for a sequence of ν values such that $\nu_1 > \nu_2 > \ldots > \nu_j > 0$. Each minimum $X(\nu_i)$ will be strictly within the constraint set. Also, with each reduction in ν the value of

$$\nu\left[\sum_{i=1}^{nc} \frac{a_i}{g_i} + \frac{a_w}{g_w} \right] \tag{3.8}$$

is reduced in influence which, in turn, forces the reduction of

$R(X)$ in the minimization of ϕ . As ν is reduced, the design $X(\nu_i)$ is allowed to approach closer and closer to the active constraints if such a move will be beneficial in reducing ϕ . Eventually, if the goal weight is equal to or greater than the true optimum weight for the problem, the residual term should be forced to zero. The process is normally not carried out that far since the analysis is considered satisfied when the condition given by Equation (3.6) is met. If it becomes apparent that the residual $R(X)$ cannot be reduced enough to meet the requirements of Equation (3.6) with further reduction of ν , then the process is terminated and the last design satisfying the analysis is considered to be an approximation of the optimum design.

If for a particular goal weight $W_{o,i}$ an analyzed design with a weight of W_i can be found, then a new goal weight is selected. A method which might be used to select this goal weight is given by the formula

$$W_{o,i+1} = (1.0 - \Delta)W_i \tag{3.9}$$

where Δ is fairly small. That is, the goal weight is set equal to some percentage of the weight of the last analyzed design. The process could be started with a fairly course value of Δ, say 0.10. When a goal is selected for which the analysis cannot be satisfied, then a smaller value of Δ could be used and the process started once again from the last analyzed design. The value of Δ could thus be refined until the reduction in weight would not be worth the effort to achieving it.

B. Development of the Function ϕ

To implement the method of synthesis outlined above, the function ϕ must be developed explicitly. To facilitate the formulation, several sets or tables of integers were constructed. Since some use will be made of these in developing ϕ , they will be defined as follows:

1. The PT_{ij} table has dimension of M by 2 where M is the number of truss members. The node numbers of the two joints which are joined by the i^{th} member are given by PT_{i1} and PT_{i2} .

2. The UU_i table has a single dimension of 3NN where NN is the total number of nodes. For a given i^{th} node, the displacement indices for the x, y, and z spacial directions are given by the value of UU_{3i-2}, UU_{3i-1} , and UU_{3i} respectively. If the node cannot have a displacement in a given direction k due to boundary onditions (i.e. the direction is supported by some external restraint), then the value of UU_k is zero.

3. The SR_{ij} table has dimensions M by 6 . It is formed by the combination of the PT_{ik} and the UU_j table. The indices of the displacements in the x, y, and z spacial directions of one of the two nodes which are

joined together by the i^{th} member are given by the values of SR_{i1}, SR_{i2}, SR_{i3} respectively. The indices of the displacement of the other node in the x, y, and z directions are similarly given by the values SR_{i4}, SR_{i5}, and SR_{i6}. Again, if a displacement in a given direction k for one of the nodes of some member i is forced to zero by an external restraint, then the value of SR_{ik} is equal to zero.

4. The DD_i table has dimension M and is used to coordinate the diameter variables. For some member i, the index j of the design variable α_j which describes the diameter of the i^{th} member is given by DD_i. If the diameter is fixed for the i^{th} member, then DD_i is given the value of zero.

5. The TT_i table has a dimension M and performs the same function for the truss members' thicknesses as the DD_i does for the members' diameters by supplying the index for the thickness design variables β_j.

After the tables have been clarified, the formulas for several useful quantities may be given. The length of the i^{th} member which is connected to the nodes p and q may be written as

$$L_i = [(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2]^{1/2} \tag{3.10}$$

where (x_j, y_j, z_j) is the coordinate of the j^{th} node.

The direction cosines of the i^{th} member, again with nodes p and q, to the x, y, and z coordinate axes are respectively given by

$$\gamma_{ix} = (x_q - x_p)/L_i$$

$$\gamma_{iy} = (y_q - y_p)/L_i \qquad (3.11)$$

$$\gamma_{iz} = (z_q - z_p)/L_i$$

The mean diameter d_i of the th member is given by

$$d_i = D_i \alpha_{DDi} \qquad (3.12)$$

where D_i is a preassigned constant such that it is either a prescribed diameter or a prescribed multiplier for the diameter parameter α_{DD_i} and is fixed throughout the entire synthesis. If it is the former possibility, then by the definition of the tables,

$$DD_i = 0$$

Thus, to obtain

$$d_i = D_i$$

requires that

$$\alpha_o = 1.0 \qquad (3.13)$$

Likewise, the thickness t_i will be given by

$$t_i = T_i \rho_{TTi} \tag{3.14}$$

which functions in the same manner as the computation of the diameter.

The equilibrium equations used for the analysis of the behavior of the truss is given by

$$\sum_{j=1}^{N} K_{\ell j} u_{jk} = P_{\ell k} \qquad \begin{array}{l} \ell = 1,2,\ldots,N \\ k = 1,2,\ldots,KT \end{array} \tag{3.15}$$

where

- N is the number of degrees of freedom per load condition
- KT is the total number of load conditions
- $K_{\ell j}$ is the force at the ℓ^{th} degree of freedom due to a unit displacement in the j^{th} node. It is called an element of the master stiffness matrix.

One method for computing $K_{\ell j}$ is from the element stiffness matrix denoted by $[K_{vb}]_i$ or $[K_{ivb}]$ for each member i of the truss. That is, an element of the master stiffness matrix may be found by the formula

$$K_{\ell j} = \sum_{i=1}^{M} \sum_{v=1}^{6} \sum_{b=1}^{6} K_{ivb}[\eta(\frac{SR_{iv}}{\ell})\ \eta(\frac{SR_{ib}}{j})] \tag{3.16a}$$

where

$$n\left(\frac{i}{j}\right) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{3.16b}$$

This is an inefficient way of assembling the master stiffness matrix if each element is formed one by one. Generally, if such a matrix is formed, it is formed at one time by considering all combinations of the indices (SR_{iv}, SR_{ib}) for each individual member. If this is done, then the method becomes very useful. In the present problem, the master stiffness matrix was never formed to conserve computer storage space.

If the direction cosines for the i^{th} member with respect to the standard coordinate system x, y, and z are γ_{ix}, γ_{iy}, and γ_{iz} respectively, then the matrix QL_i shall be defined as

$$QL_i = \begin{bmatrix} \gamma_{ix}^2 & \gamma_{ix}\gamma_{iy} & \gamma_{ix}\gamma_{iz} \\ \gamma_{iy}\gamma_{ix} & \gamma_{iy}^2 & \gamma_{iy}\gamma_{iz} \\ \gamma_{iz}\gamma_{ix} & \gamma_{iz}\gamma_{iy} & \gamma_{iz}^2 \end{bmatrix} \tag{3.17}$$

Now the local stiffness matrix in the standard coordinate system may be written in the form of the partitioned matrix

$$[K_{ivb}] = \frac{A_i E_i}{L_i} \left[\begin{array}{c|c} QL_i & -QL_i \\ \hline -QL_i & QL_i \end{array}\right] \tag{3.18}$$

where

A_i is the area of the member or $\pi d_i t_i$

E_i is Young's modulus for the member

In forming the value of the residual R as described in Chapter 3, the residuals $r_{\ell k}$ are first computed. This is written as

$$r_{\ell k} = \sum_{j=1}^{n} K_{\ell j} u_{jk} - P_{\ell k} \qquad \begin{aligned} \ell &= 1,2,\ldots,N \\ k &= 1,2,\ldots,KT \end{aligned} \tag{3.19}$$

Now R can be given by

$$R = \sum_{k=1}^{KT} \sum_{\ell=1}^{N} r_{\ell k}^2$$

If a local force element for member i is denoted by $\tilde{r}_{vk}^i$, then

$$\tilde{r}_{vk}^i = \sum_{b=1}^{6} [\tilde{K}_{ivb} \, u_{SR_{ib},k}] \tag{3.20}$$

The residual $r_{\ell k}$ for a given load condition k and degree of freedom ℓ is found by

$$r_{\ell k} = \sum_{i=1}^{M} \sum_{v=1}^{6} \left[\tilde{r}_{vk}^i \, n \left(\frac{SR_{iv}}{\ell} \right) \right] - P_{\ell k} \tag{3.21}$$

Again it is unlikely that each of the residuals would be computed one by one, but rather, like the master stiffness matrix, all residuals resulting from one load condition would be formed at one time. By noting the form of the partitioned matrix in Equation (3.18), considerable computational efforts can be saved in forming the residuals. The above formulation of R completes part of the function ϕ as given by

$$\phi = R + \nu[\sum_{i=1}^{nc} \frac{a_i}{g_i} + \frac{a_w}{g_w}]$$

In the next part, attention is turned to the constraints g_i and the weight constraint g_w .

The first constraints to be considered will be those of stress. The computation of the stress in the i^{th} member with nodes p and q from the k^{th} loading is given by

$$\sigma_{ik} = \frac{E_i}{L_i} \Big| \gamma_{ix}(u_{px,k} - u_{qx,k}) + \gamma_{iy}(u_{py,k} - u_{qy,k}) + \gamma_{iz}(u_{pz,k} - u_{qz,k}) \Big| \tag{3.22}$$

where $u_{jc,k}$ is the displacement at the j^{th} node in the direction of the c axis for the k^{th} load condition. The values of the crippling and Euler buckling of a thin wall tubular member as a function of the member's mean diameter d_i and thickness t_i are respectively given by

$$(\sigma_{cr})_i = - \frac{k_2 E_i t_i}{d_i}$$

$$(\sigma_b)_i = - \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \tag{3.23}$$

where k_2 is a material constant.

Listing the constraints which must be satisfied in each member for every load condition gives the constraints, as specified earlier,

$$\begin{aligned} g_1 &= \sigma_{yt} - \sigma_{ik} > 0 \\ g_2 &= \sigma_{ik} - \sigma_{yc} > 0 \\ g_3 &= \sigma_{ik} - (\sigma_{cr})_i > 0 \\ g_4 &= \sigma_{ik} - (\sigma_b)_i > 0 \end{aligned} \tag{3.24}$$

Without further ado, the contribution to ϕ due to the stress constraints is

$$\phi_\sigma = \sum_{k=1}^{KT} \sum_{i=1}^{M} K_d \left[\frac{1.0}{\sigma_{yt} - \sigma_{ik}} + \frac{1.0}{\sigma_{ik} - \sigma_{yc}} + \frac{1.0}{\sigma_{ik} - (\sigma_{cr})_i} + \frac{1.0}{C_{z1}(\sigma_{ik} - (\sigma_b)_i)} \right] > 0 \tag{3.25}$$

where K_d and C_{z1} are nondimensionalizing factors.

The constraints which were placed on the displacements in the specification of the problem were

$$\begin{aligned} g_5 &= u_{up} - u_{jk} > 0 \\ g_6 &= u_{jk} - u_{low} > 0 \end{aligned} \tag{3.26}$$

where u_{up} and u_{low} are the upper and lower limits on the displacements. The effect of the function due to these constraints is given by

$$\phi_d = \sum_{k=1}^{KT} \sum_{j=1}^{N} K_{uz} \left[\frac{1.0}{u_{up} - u_{jk}} + \frac{1.0}{u_{jk} - u_{low}} \right] \tag{3.27}$$

where K_{uz} is a nondimensionalizing constant.

Side constraints in this problem have been imposed on the truss members. These constraints may be written as

$$\begin{aligned} g_7 &= d_{max} - d_i > 0 \\ g_8 &= t_{max} - t_i > 0 \\ g_9 &= d_i > 0 \\ g_{10} &= r_u - d_i/t_i > 0 \\ g_{11} &= d_i/t_i - r_\ell > 0 \end{aligned} \tag{3.28}$$

The total effect on ϕ from all of these constraints is

$$\phi_s = \sum_{i=1}^{M} \left[\frac{1.0}{d_{max} - d_i} + \frac{1.0}{t_{max} - t_i} + \frac{1.0}{d_i} + \frac{1.0}{r_u - d_i/t_i} + \frac{1.0}{d_i/t_i - r_\ell} \right] \tag{3.29}$$

Finally, consideration must be given to the weight constraint. The weight is given by

$$W = \sum_{i=1}^{M} wf_i \alpha DD_i \beta TT_i \tag{3.30}$$

where

$$W\bar{T}_i = \pi\rho D_i T_i$$

$$\rho = \text{unit weight of material}$$

Since the constraint may be written as

$$g_W = W_o - W > 0$$

then the contribution to the function ϕ is

$$\phi_W = \frac{W_m W_o}{W_o - W} \tag{3.31}$$

where

W_o is used as the goal weight and as a non-dimensionalizing factor.

W_m is used to vary the influence of ϕ_W with respect to ϕ .

Once all of the constraints are considered, the function ϕ may be written first as

$$\phi = R + \nu(\phi_\sigma + \phi_d + \phi_s + \phi_W) \tag{3.32}$$

and then in its full form

$$\phi(X) = \sum_{k=1}^{KT} \sum_{\ell=1}^{N} r_{\ell k}^2$$

$$+ \nu \left\{ K_d \sum_{k=1}^{KT} \sum_{i=1}^{M} \left[\frac{1.0}{\sigma_{yt} - \sigma_{ik}} + \frac{1.0}{\sigma_{ik} - \sigma_{yc}} \right.\right.$$

$$\left. + \frac{1.0}{C_{z1}(\sigma_{ik} - (\sigma_b)_i)} + \frac{1.0}{\sigma_{ik} - (\sigma_{cr})_i} \right]$$

$$+ K_{uz} \sum_{k=1}^{KT} \sum_{j=1}^{N} \left[\frac{1.0}{u_{up} - u_{ij}} + \frac{1.0}{u_{ij} - u_{low}} \right]$$

$$+ \sum_{i=1}^{M} \left[\frac{1.0}{r_u - d_i/t_i} + \frac{1.0}{d_i/t_i - r_\ell} + \frac{1.0}{d_i} \right.$$

$$\left. + \frac{1.0}{d_{max} - d_i} + \frac{1.0}{t_{max} - t_i} \right]$$

$$\left. + \frac{W_o W_m}{W_o - W} \right\} \qquad (3.33)$$

The gradient of the function will be needed in the minimization process. With the function in hand, the gradient is found by some tedious but straight forward operations. The results of such operations are presented in Appendix A.

CHAPTER IV

OPERATIONAL TECHNIQUES

In order to handle realistic problems, the approach to synthesis outlined in the previous chapter was programmed for the computer. To make the formulation operational, certain procedures or devices were incorporated in the final program. Those which are discussed in this chapter are the minimization scheme and a scaling transformation.

A. Minimization Scheme

Central to this method of synthesis is the minimization of the function ϕ . Since the function is parametrically dependent on the values of ν and the goal weight W_0 , a new minimization is started every time ν or W_0 is modified. Occasionally, two or three minimizations are required to locate one analyzed design for a given goal weight.

The possible techniques used for these minimizations are the same as those associated with unconstrained minimization problems. Presented here are two important methods of this type. They are the Fletcher-Reeves method of conjugate gradients and the Fletcher-Powell method (see References 11 and 12). The relative merits and drawbacks of each will be discussed and the reasons for the selection of the Fletcher-Reeves method to obtain numerical results for the test problems will be given. Before the descriptions of the methods mentioned above are presented, some of the considerations which should be made in selecting a minimization procedure will be set forth.

Due to the fact that there are a number of minimizations performed in this synthesis approach, it seems natural that an efficient minimization scheme would be desirable. By efficient, it is meant that the scheme should be able to find the minimum of the function in a relatively small number of operations. This need is even more strongly emphasized since practical problems have a large number of variables. Even for efficient schemes, the amount of work is usually dependent in some degree on the number of variables in the problem. A second important consideration is the ability to handle a large number of variables in terms of computer storage. For a moderately complex truss system with several load conditions the number of variables could become rather large. After all factors have been considered, it may become obvious that the selection of a method will require compromises in the goals mentioned above.

The minimization of a continuous function f with continuous first and second partial derivatives will be considered. The function ϕ satisfies such conditions except where the constraints are exactly critical. Since the region where the minimization search is taking place is strictly within the constraints the methods which can be used for minimizing f are also applicable for ϕ .

From the standard quadratic expansion about the point $\vec{x}^0$

given by

$$f = f^{o} + \sum_{i=1}^{n} a_i(x_i - x_i^{o}) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} G_{ij}(x_i - x_i^{o})(x_j - x_j^{o}) \tag{4.1}$$

where

$$G_{ij} = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)_{\overline{x}^{o}}$$

$$a_j = \left(\frac{\partial f}{\partial x_j}\right)_{\overline{x}^{o}}$$

the vector between the point $\overline{x}$ and the minimum $\overline{x}^*$ can be shown to be

$$\overline{x}^* - \overline{x} = -G^{-1}\nabla f(\overline{x}) \tag{4.2}$$

To approximate the matrix G^{-1} in Equation (4.2), an n by n matrix H is employed. Initially chosen to be positive definite, H is modified in a manner so that it tends to G^{-1} as the minimization process is accomplished. The algorithm for the Fletcher Powell method is

Select H_o as any arbitrary positive definite matrix; usually the identity matrix.

$\overline{x}_o$ as an initial point.

then

$$\bar{s}_i = -H_i \nabla f(\bar{x}_i)$$

f_i = minimum of $f(\bar{x}_i + \alpha_i \bar{s}_i)$ with respect to the variable α_i occurring at α_i^*

$$\bar{\sigma}_i = \alpha_i^* \bar{s}_i$$

$$\bar{x}_{i+1} = \bar{x}_i + \bar{\sigma}_i$$

$$H_{i+1} = H_i + A_i + B_i$$

where

$$A_i = \frac{\bar{\sigma}_i \bar{\sigma}_i^T}{\bar{\sigma}_i^T \bar{y}_i}$$

$$B_i = \frac{-H_i \bar{y}_i \bar{y}_i^T H_i}{\bar{y}_i^T H_i \bar{y}_i}$$

$$\bar{y}_i = \nabla f(\bar{x}_{i+1}) - \nabla f(\bar{x}_i)$$

Fletcher and Powell showed that if certain precautions are taken, H will always remain positive definite. This guarantees that the method will converge since the function f initially decreases along the direction $\bar{s}_i$ causing the function to be decreased for each iteration of the process. Also, for a quadratic function of n variables, it was shown that after n or fewer steps, the minimum will be reached and H will converge to the inverse of the matrix of second partial derivatives.

The Fletcher-Reeves method of conjugate gradients is one in which a direction for minimization $\bar{s}_i$ is generated from the gradient and the previous direction $\bar{s}_{i-1}$. Specifically, the algorithm for the routine is as follows:

Choose $\quad \bar{x}_0 \;=\;$ an arbitrary initial point

with $\quad \bar{s}_0 \;=\; \nabla f(\bar{x}_0)$

then

$$\bar{x}_{i+1} \;=\; \bar{x}_i \quad h^*\bar{s}_i \quad \text{where } h^* \text{ is such that}$$

$$f(h) = f(\bar{x}_i - h\bar{s}_i) \text{ is minimized at}$$

$$h = h^* \;.$$

$$\bar{g}_{i+1} \;=\; \nabla f(\bar{x}_{i+1})$$

$$\beta_i \;=\; \bar{g}^2_{i+1}/\bar{g}^2_i$$

$$\bar{s}_{i+1} \;=\; \bar{g}_{i+1} + \beta_i\bar{s}_i$$

The Fletcher-Reeves method is also guaranteed, except for effects due to rounding errors, to find a minimum of any quadratic function of n variables in n or fewer steps.

Even though for the quadratic function both methods can theoretically achieve a minimum in the same number of steps, the Fletcher-Powell method is generally more powerful. For ill behaved functions, seemingly the natural condition for real problems, the Fletcher-Powell method does produce a minimum more efficiently than the Fletcher-Reeves method. Unfortunately, the Fletcher-Powell method requires the storage and manipulation of an n by n matrix. This is undesirable since the storage of very large matrices requires highly time consuming operations such as reading and writing on drums and tapes. The Fletcher-Reeves method, however, requires the storage of only a few n vectors.

Despite the fact that the Fletcher-Reeves method is less powerful, it was chosen as the minimization scheme to use in the study of this synthesis approach. The loss of efficiency in minimization was felt to be off set by an increase in capability for handling large problems.

B. Scaling Transformation

In the first tests of this synthesis approach, it was found that the function ϕ was often difficult to minimize using the Fletcher-Reeves method. After three or four steps, the process would produce very little reduction of the function value and small changes in the design vector. Even after $3n/2$ or $2n$ steps, where n was the total number of synthesis variables, an insignificant amount of progress would be made. From the elements of the gradient, it was clear that the design vector was not at or even near the minimum. In essence, many of the characteristics of an ill conditioned function were present.

In an effort to improve the conditioning of the function, a scaling transformation was considered. Such an operation is a multiplication of the variables by constants and can be represented by the matrix formulas

$$Y = D^{-1}X \tag{4.3a}$$

$$X = DY \tag{4.3b}$$

where

X is the n vector in the original system.

Y is the n vector in the transformed system.

D is an n by n nonsingular diagonal matrix which contains the multiplying constants.

The ability of such a transformation to improve the conditioning of a function can be explained in terms of eigenvalues and the Gerschgorin Theorem.

Gerschgorin's Theorem

Every eigenvalue of the matrix A lies in at least one of the disks centered at a_{ii} and of radii

$$R_i \equiv \sum_{i \neq j} |a_{ij}| \tag{4.4}$$

It will become clear that a characteristic of an effective transformation is its ability to reduce the range in which the eigenvalues of the matrix A lie.

Consider the quadratic function f given by

$$f = \frac{1}{2} X^T A X \tag{4.5a}$$

where

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = a_{ij} \tag{4.5b}$$

is the form of its second partial derivatives. If f is assigned a value and if A is positive definite, then the locus of points generated by Equation (4.5a) is a.. ellipoisal hypersurface. The ratio of the principal axes are proportional to the square root of the ratio of the eigenvalues obtained from A (i.e. the matrix of second partial derivatives). If all the eigenvalues are equal, the hypersurface is a sphere. For this special case of all the eigenvalues of A being equal, the function is very easy to minimize. Indeed, the conditioning of a function for minimization is usually related to the conditioning number C_N defined as the ratio of the largest to least eigenvalue or

$$C_N = \lambda_{max}/\lambda_{min} \tag{4.6}$$

and the least value of C_N is 1.0 which occurs when all the eigenvalues are equal.

The function f of Equation (4.5) can be written in terms of the transformed variables as

$$f = \frac{1}{2} Y^T \tilde{A} Y \tag{4.7}$$

where

$$\tilde{A} = D^T A D \tag{4.8}$$

One selection for the transformation constants may be given by

$$d_{ii} = 1/(c\sqrt{a_{ii}})$$

$$d_{ij} = 0 \quad i \neq j \tag{4.9}$$

If this transformation is used, all the Gerschgorin disks will be centered about the point $1/c^2$. In Reference 10, it has been shown that the eigenvalues of a symmetric, positive definite system are bounded above by $\lambda_{max} \leq n\tilde{a}_{ii}$ where n is the maximum number of nonzero elements in any row.

One point which should be emphasized is that the scaling given by Equation (4.3) is generally unable to produce an "optimum" transformation. An optimum transformation is one which reduces C_N to its minimum value. The conditioning of some functions can actually be made worse by a transformation of the form given by Equation (4.3). However, a scaling of this type is often useful in achieving a meaningful improvement in the conditioning of the function (see Reference 10).

To illustrate the effects of the transformation on a two dimensional ellipsoid, the curves for $f = 100$ in the original and scaled coordinates are shown in Figure 3 for the equations

$$f = \frac{1}{2} X^T \begin{bmatrix} 8 & 12 \\ 12 & 50 \end{bmatrix} X$$

$$= \frac{1}{2} Y^T \begin{bmatrix} 1.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix} Y$$

$$= 100$$

where

$$X = DY$$

$$D = \begin{bmatrix} \sqrt{2}/4 & 0.0 \\ 0.0 & \sqrt{2}/10 \end{bmatrix}$$

Even though the principal axes of the ellipsoid f does not coincide with the axes of the coordinate system, the reduction of the eccentricity can be detected by visual inspection. The conditioning numbers for the original and transformed systems are 8.9 and 4.0 respectively.

So far in this discussion, the application of the scaling transformation has been made only to the well defined case of quadratic functions which describe ellipsoids in n space. What of the function ϕ which is not quadratic? First, a portion of some ellipsoid can be approximately fitted onto the contour surface of the function ϕ in the region of any given point. If it is assumed that the function and the ellipsoid are similar in that region, then local scaling factors for ϕ can be computed from Equation 4.9. These factors should at least improve the conditioning of the function in the local region of the point. As the region of the minimum is approached, it would be hoped that the scaling factors would converge to constant values. That is, in the vicinity of the minimum, the function should be closely approximated by the quadratic form of the Taylor series.

To investigate the effect of scaling on the function ϕ, the three bar truss problem with only the stress constraints were studied in some detail. To accomplish this, the matrix of second partial derivatives was formed. Using this matrix, eigenvalues were found for several points. The results for two points are presented in Table 1. One point was an arbitrary, acceptable design and the other point was at the minimum of the function for a given ν and W_0. As it can be seen, the scaling significantly reduced the value of C_N for both points.

Since the conditioning numbers for local regions of the function were reduced, it would appear that the minimization scheme might be able to work fairly effectively. This proved to be true. Using scaling, the minimization of the function with the Fletcher-Reeves method was able to proceed in a satisfactory manner.

One question which arose in dealing with the scaling factors was that of when to rescale or, at least, to check to see if rescaling was necessary. Since it was assumed that a scaling was only valid for the local region of the point at which it was done, it was apparent that rescaling would probably have to be redone occasionally. To do it seldom would invite the possibility that, after three or four Fletcher-Reeves moves, the process would bog down until the next rescaling. This would be little better than the initial problem. Yet, to rescale too frequently would detract from the power of the Fletcher-Reeves scheme near the minimum since the direction vector $\bar{s}$ should be replaced by the gradient after such a rescaling.

After running several tests with rescaling at various intervals, the occurance of both problems was verified. A sequence of function values for rescaling every twenty steps for a thirty variable problem is shown in Figure 4. The first two scalings produce rapid decreases in the function value for the gradient moves. The third rescaling, done near the minimum, did not produce any great movement. In Figure 5, the effect of rescaling every seven iterations have been plotted for the same problem. In the latter stages of the process, it appears that the Fletcher-Reeves method is slowed by the rescaling. These results indicated that rescaling should be done often at the start of the minimization and less often towards the end of the process.

To accomplish these requirements, the following algorithm was developed. To begin the description of the algorithm, several variables should be defined. The variable C_b was the number of Fletcher-Reeves steps that the minimization was allowed to take before the scaling factors were checked and changed if necessary. The counter C_a indicated the number of Fletcher-Reeves iterations since the last time the scaling factors were checked and C_d counted the number of times the scaling factors were checked and changed for some value of C_b . The flow diagram of the logic is shown in Figure 6. Briefly, C_b was set equal to one at the beginning of the minimization. Whenever C_a equaled C_b , the scaling factors were checked. If

$$\left| \frac{d_{ii,new} - d_{ii,old}}{d_{ii,old}} \right| < \Delta_c \qquad i = 1,2,\ldots,n \qquad (4.10)$$

was satisfied, then the old value of the scaling factors were retained, C_b was doubled, C_d was set to zero, and the next Fletcher-Reeves step was executed. If the condition given by Equation (4.10) was not satisfied, then the scaling factors were changed, C_d was incremented by one and the $\overline{s}$ vector was replaced by the gradient. In either case, C_a was zeroed. Also, to reduce the chances of oscillation in the values of the scaling factors, C_b was increased by one if C_d managed to become equal to three. Such oscillations were observed to occur before the inclusion of this guard. The value of Δ_c used in this study was 0.60.

CHAPTER V

RESULTS

A program incorporating the synthesis approach previously described was written and us... to obtain numerical results for several test problems. A listing of the program can be found in Appendix B and the results for the test problems are summarized in this chapter.

The program was written in ALGOL 60 and was run on a UNIVAC 1107. The running times are presented in Table 2. Two times are reported for each test. The first time is the number of seconds which elapsed between the analysis of the initial design and the synthesis of the last analyzed design. The second is the time required for the program to realize that the goal weight was less than the optimum weight after the last analyzed design was found. It is estimated that these running times would be reduced by a factor of 4 to 7 if the programming were to be done in FORTRAN. This is a standard estimation based on comparisons of programs coded in both ALGOL and FORTRAN.

The times were presented to give some basis of the evaluation of the method's efficiency. However, caution should be used when drawing conclusions from these times.

The test problems included planar and three dimensional examples. For Case 1, a three bar truss, the optimum design was given in Reference 13. Also, results from References 1 and 8 of

Cases 2.A, 2.B, 3, and 4.A were available for comparison with the results obtained by this algorithm. The run times and final weights for both sets of results are given in Tables 15 and 16 respectively.

A. Planar Case

1. Three Bar Truss

Consider a three bar truss as shown in Figure 1 subjected to the two independent load conditions P_1 and P_2. If the constraints described in Chapter I are imposed and a high value of Young's modulus is selected, then the problem essentially becomes the same as an example presented in Reference 13 which has only two constraints

$$\sigma_{yt} - \sigma_i > 0$$

$$\sigma_i - \sigma_{yc} > 0$$

where

$$\sigma_{yt} = 20000 \text{ psi},$$

$$\sigma_{yc} = -15000 \text{ psi}$$

For this problem the optimum weight is $W_{opt} = 79.2\rho$ which is given by the design $A_1 = A_3 = 0.788 \text{ in}^2$ and $A_2 = 0.41 \text{ in}^2$. If the unit weight of the material is chosen to be 1.0 lb/in^3, then the optimum weight is 79.2 lb. Also, with the thickness preassigned as $t_i = 1/4\pi$ in. the diameter will be $d_1 = d_3 = 3.15$ in. and

d_2 = 1.64 in. at the optimum design.

To assure that the displacement limits would not be active, the limits were given the large values of ± 1.0 in. Since displacements this large might violate the assumption of small displacements made in the formulation of the analysis, the displacements of the final design were checked. The largest displacement was less than 0.003 in. for an assumed value of 3.0×10^8 psi for Young's modulus E. Hence, the validity of the assumption of small displacements was maintained.

The goal weight was selected to be 5% less than the weight of the last analyzed design. As can be seen from Table 3, the synthesis method was able to approximate the optimum within the expected amount of 5%. One interesting aspect of this problem is that since there are only two member design variables d_1 and d_2, i.e. $d_3 = d_1$, the projection of the path of all the design variables can be made on a plane, i.e. the d_1, d_2 plane. The path for Case 1 is shown in Figure 7. Designs which satisfy the analysis according to Equation (3.6) are denoted with circles while designs where minimum of the function ϕ occurred and ν was reduced are marked with triangles. Designs B and D were found by a proportional decrease in member diameters with an appropriate increase in the displacements. When such a modification could not be made so that the constraints were satisfied, the synthesis process was used to try to locate an analyzed design for the new goal weight. Design C is the result of the synthesis algorithm. Design E is also the result of the process even though no analyzed design could be found. The synthesis was terminated after three minima were found for three decreasing values of ν and a goal weight of 76.1 lb. That the analysis could not be satisfied became evident from an extrapolation for $\nu = 0$ of the values of the residual R at these minima.

Figure 7 would seem to indicate that some of the intermediate designs between B and C did not satisfy the constraints. However, it must be remembered that the constraints shown in the figure are merely projections onto the d_1, d_2 plane and must be satisfied only for analyzed designs. For example, any d_1 and d_2 which are positive would satisfy the stress and displacement constraints if all the displacements were set equal to zero. The design certainly would not be accurately analyzed but would satisfy the behavior constraints since all the member stresses would be zero.

2. Eleven Bar Truss

Consider the eleven bar truss shown in Figure 8 which is to be constructed of steel. The material properties of the steel are $E = 3 \times 10^7$ psi and $\rho = 0.28$ lb/in^3. The three loading conditions to be applied to the structure are listed in Table 4. The stress limits include Euler buckling, crippling, tensile, and compressive yield stresses. The yield strength of the material was assumed to be ± 50000 psi. The design variables used in this study were the member diameters. The thickness of all the tubular members were set to a value of $1/10\pi$ or 0.0318 in.

Since the reactions are not symmetric, the design itself will not necessarily be symmetric despite the symmetry of the load conditions. Hence, linking of the design variables was not used in this study.

As in all other tests of this synthesis algorithm, the goal weight was selected to be 5% less than the weight of the last analyzed design. This was considered a reasonable reduction. After the final design has been produced using this reduction, the synthesis could be restarted at this last analyzed design for the finer tolerances of 1.0% or even 0.1% if the need presented itself.

The results for the displacement limits of ± 0.15 in. and ± 0.25 in. are presented in Tables 5 and 6 respectively. For the displacement limits of ± 0.15 in., Case 2.A, the design is limited by the displacement constraint in the downward direction at node 2 for the first load condition. The short time shown in Table 2 required to locate the final design is due to the fact that the final design was found from the initial, analyzed design by a simple proportional reduction of the design variables with the appropriate adjustments for the displacements. In this case the synthesis process was only used to determine that the weight of the design given by the scaling was at least within 5% of the optimum weight.

The design with displacement limits of ± 0.25 in., Case 2.B, was essentially fully stressed. The final weight was 57.6 lb and the last goal weight was 54.7 lb. As indicated in Table 16, the optimum weight is near 55.5 lb. It was felt that the long time required to terminate the synthesis for this case was due to the fact that the goal weight was near the optimum.

From Table 16, the final weights given by this algorithm agree with the results given in Reference 1. However, for Case 2.A the values of the diameters are considerably different for the two designs. This could be expected since it has been pointed out by Reference 1 that for displacement limited problems, a number of designs of equal weight can be found so that all have the same value for a particular displacement. The designs for Case 2.B are fairly similar.

There would appear to be some reduction in the run times for the new algorithm. Without the termination time for the results of Reference 1, the over all efficiency of the methods can only be qualitatively stated. Beside being different in formulation, the two approaches were carried out with different minimization schemes.

B. Three Dimensional Cases

1. Nine Bar Truss

Consider a nine bar space truss shown in Figure 9 subjected to the two load conditions given by Table 7. The material from which the structure is to be made is magnesium with $E = 6.5 \times 10^6$ psi and $\rho = 0.065$ lb/in^3. The tensile and compressive yield of the material were 30000 psi and -17000 psi respectively. Both the thicknesses and diameters were allowed to vary and linking was imposed so that the groups d_1; t_1; $d_2 = d_3 = d_4 = d_5$; $t_2 = t_3 = t_4 = t_5$; $d_6 = d_7 = d_8 = d_9$; and $t_6 = t_7 = t_8 = t_9$ were formed to give a total of six member design variables.

Starting from the initial design weighing 10.92 lb, the final design for this structure, Case 3, was 5.97 lb for a displacement limit of ± 10.0 in. Again this limit was made artificially large so that the stress constraints would control the design. To assure that the assumption of small displacements was valid, the maximum displacement of the final design was checked and found to be less than 0.25 in. -- certainly small enough to satisfy the assumption. The sizes of the initial and final diameters and thicknesses for this problem are given in Table 8.

Again, compariable results are available for this test case. From Table 8, the final weights are quite close and the designs are almost the same. In this problem, the run time using the new algorithm was clearly good compared to that of the previous method.

2. Twenty Five Bar Truss

Consider the truss shown in Figure 10 which is to be constructed of aluminum with $E = 10^7$ psi and $\rho = 0.1$ lb/in^3. The two load conditions are presented in Table 9. The material yield stress was taken to be ± 40000 psi. As usual the side constraints were placed on the member sizes but, notably, an upper limit of 100.0 was placed on the diameter to thickness ratio. The linking feature was employed so that the following groups of diameters were formed: d_1; $d_2 = d_3 = d_4 = d_5$; $d_6 = d_7 = d_8 = d_9$; $d_{10} = d_{11}$; $d_{12} = d_{13}$; $d_{14} = d_{15} = d_{16} = d_{17}$; $d_{18} = d_{19} = d_{20} = d_{21}$; and $d_{22} = d_{23} = d_{24} = d_{25}$. The thicknesses were linked in a similar manner.

For Case 4.A the displacement limit was set at ± 0.35 in. The initial and final designs are presented in Table 10. The final design, which weighted 562.9 lb, was limited by the y displacement component of nodes 1 and 2 for both the load conditions. Also, the Euler buckling constraint for members 19 and 20 appeared to be critical for the second loading condition.

Comparison of design evolved by the new algorithm and the results given by Reference 8 shows a marked difference in the value of the diameters and thicknesses but a close agreement in the final weights. Again this is primarily due to the fact that the design is displacement limited. The run times are quite different with the new algorithm being the slower of the two. Part of the explanation is that the results for this problem presented in

Reference 8 were obtained using a variable metric minimization scheme. Such a scheme can be expected to complete the problem in about a third of the time required by the modified steepest descent minimization scheme used to obtain the results for Case 2.A, 2.B, and 3 in Reference 1.

The displacement limits were essentially removed by relaxing them to 10.0 in. and the synthesis for Case 4.B was initiated from a design weighing 581.3 lb, slightly heavier than the final design for Case 4.A. The final weight of 285.6 lb and other results for the case are given in Table 11. The active constraint was the upper limit of the diameter to thickness ratio for members 12 through 25. Also, the Euler buckling constraint was active for members 16 and 25 for the first load condition and members 19 and 20 for the second load condition. The magnitude of the greatest displacement is 0.805 in. for the y displacement at nodes 1 and 2 for the first load condition. From Table 11, the observation was made that members 10 through 13 tended to become small.

Starting from the final design of Case 4.B weighing 285.6 lb a final weight of 202.3 lb was obtained when the upper limit of the diameter to thickness ratio was raised to 400 for Case 4.C. The final design, given by Table 12, was stress limited in that at least one member of each group was bounded by the Euler buckling and crippling limits except for member 1. Member 1 was in tension for both loadings. The y displacement of nodes 1 and 2 gave the greatest displacement of 1.05 in. under the first load condition.

Again, there appears to be an attempt to eliminate members 10 through 13.

3. Thirty Bar Truss

Consider the dome shaped truss shown in Figure 11 to be made from steel with $E = 30 \times 10^6$ psi and $\rho = 0.28$ lb/in^3 to be Case 5. The yield strength of the material was taken to be ± 40000 psi. The two loading conditions imposed on the structure are given by Table 13. Linking of the thicknesses of members 1 through 6, 7 through 12, 13 through 18, and, finally, 19 through 30 was imposed. A similar design variable linking was made for the diameters.

A displacement limit of ± 1.60 in. was placed on the design. Starting from a design weighing 266600 lb a final design with a weight of 199700 lb was synthesized. The dimensions for the diameters and thicknesses for this problem are given in Table 14. The active constraints were the downward components of the displacements of nodes 4, 5, and 6 under the second load condition.

CHAPTER VI

DISCUSSION AND CONCLUSION

An algorithm was presented for finding the minimum weight design of a general, linear truss subject to a set of behavior and side constraints under an arbitrary number of imposed load conditions. This was accomplished by defining a goal weight and then minimizing the residuals subject to the original and weight constraints. Then, the Fiacco McCormick technique was used to transform this constrained minimization into one without constraints.

The ability of this algorithm to minimize the weight of the general truss was tested by considering several problems which have previously been solved using other algorithms. The results from these tests agreed with those from the other methods. From a comparison of run times, not including the termination times, it would appear that the new algorithm is usually faster in achieving a final design than similar methods. A comparison of the termination time, i.e. the time elapsed from the final design to the termination of the process, was not possible.

The formulation of ϕ using the Fiacco McCormick method provided one clear advantage. Namely, the weight of the analyzed design produced by the synthesis was often considerably lower than the goal weight if the goal weight was much greater than the optimum. For the original algorithm which was developed within framework of the integrated approach, the design weight was usually very close to the goal weight. In the formulation of that method, the problem was transformed into an unconstrained function by using an external penalty function which had a value only if a constraint was violated. However, since the Fiacco McCormick formulation usually took longer to find an analyzed design, only part of this advantage could be realized.

Some difficulties arose in the operation of the algorithm. These may generally be divided into two main areas. The first is concerned with the modification of an analyzed design to satisfy a new, lowered goal weight. The second is concerned with the long termination time. Hopefully, the following discussion can suggest some useful solutions to these difficulties.

Presently, an analyzed design is proportionally reduced so that all members experience the same percent reduction in their cross sectional area. Such a modification allows a very simple procedure to find the analysis of the design if all members are allowed to vary and no side constraints are active. Even though this modification has been found to be useful for the initial designs used for the test cases, it was evident that this method rarely provided any advantage in the latter stages of the synthesis. Another approach to the modification may be to extrapolate a new, modified design from the previous two analyzed design. This method would tend to account for the local shape of the constraint surfaces. Even though the advantage of possibly obtaining another analyzed design from the current design is lost, the modified design from the proposed method may provide a better initial point in minimization of the function ϕ and hence may have a tendency to shorten the run time.

The second major problem is the long termination time, (i.e. the time required for the program to realize that a goal weight is less than the optimum weight). Since the termination requires a succession of minimization for a sequence of ν, this time depends greatly on the time to minimize ϕ.

To reduce the time of the minimization, certain areas might be considered. First, an extrapolation of the minima have been developed for the Fiacco McCormick technique. Essentially, the extrapolation provides for an estimation of the minimum for a given ν based on the minima for the previous two values of ν. Even though the estimation seldom locates the minimum exactly, it usually will provide a significantly better starting point than the minimum for the previous ν.

A second possibility in reducing the time for minimization is the selection of the criterion for the minimum. This difficulty is not unique to this method but the nature of this formulation does provide special considerations. Using the standard tests for the gradient and the change in the function value, the question of how stringently the criteria should be applied arises. It is felt that these criteria can be applied fairly loosely. Since a new minimization of ϕ will be reinitiated after a minimum is found and ν is reduced, the extra effort required to find the minimum very accurately seems somewhat wasteful. The criteria does need to be harsh enough that the residual R will eventually be sufficiently reduced if the goal weight is greater than the optimum weight. Also, it would be desirable that a minimum will be accurate enough to serve as an adequate starting point for the minimization of ϕ for the reduced value of ν.

The formulation of the function ϕ with a constrained goal weight suggests or allows the development of additional algorithms. One of these algorithms can be used to determine the feasibility of using particular configuration of a truss as a solution to a given structural problem. Generally, along with the loads and constraints, the feasibility study specifies a weight for which the truss may not be greater. Then the problem is one to find a design which satisfies the constraints and weighs less than the specified weight -- precisely the same problem which is solved over and over by the present algorithm. To incorporate this algorithm, the goal weight is replaced by the specified weight and a set of design variables are selected so as to satisfy the weight and side constraints. Then the displacements are set to zero, which also means that the stresses are zero, so that all the behavior constraints are satisfied. Finally, the process is initiated. If the truss is a feasible structural form for the problem, a design will be given.

Otherwise, the process will indicate that the particular configuration of the truss is an unfeasible solution.

A second additional algorithm which might be developed is one to find the optimum design by selecting a goal weight less than the optimum. Started in the same manner as the algorithm above, the process should indicate that the goal weight is less than the optimum weight. As a goal weight is shown to be unacceptable, a new, higher goal weight is selected until an analyzed design is found. Since Table 2 indicates that demonstrating a goal weight is unattainable takes considerable time, an effective form of an extrapolation would be required for the method to be efficient. An algorithm more efficient than the one presented by this paper might be developed if such an extrapolation could be found.

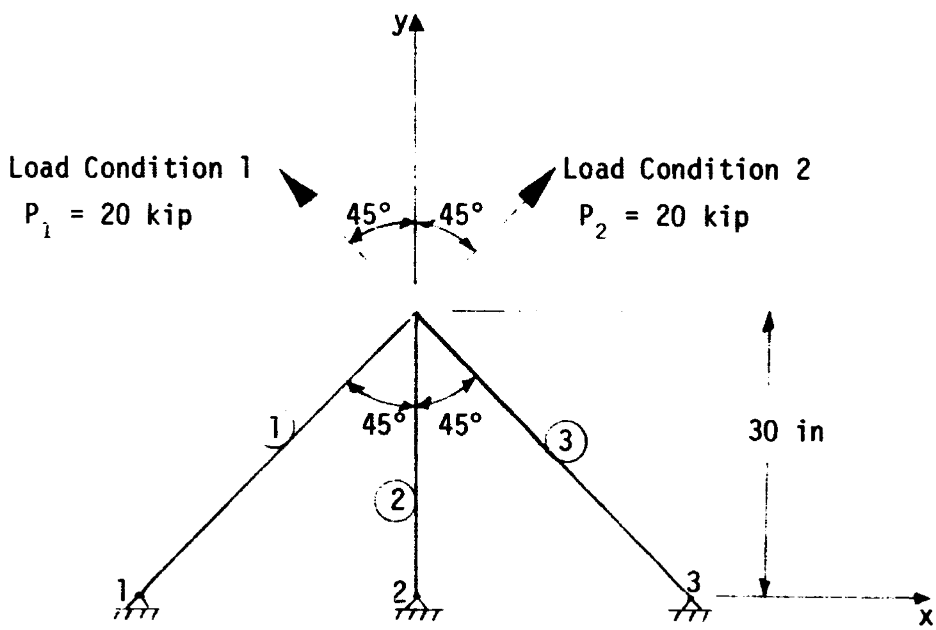

Figure 1 Three Bar Truss

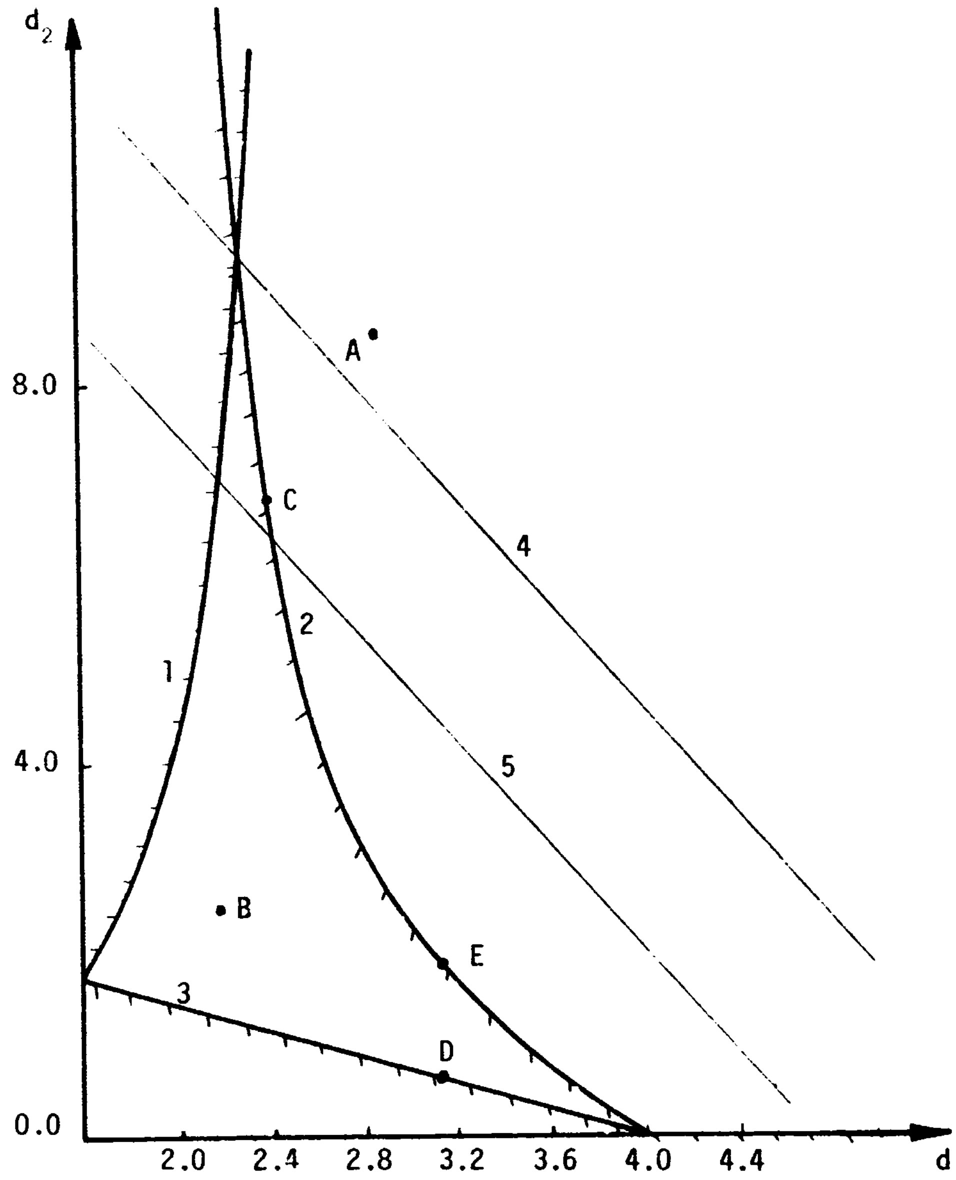

Figure 2 Design Space for Three Bar Truss

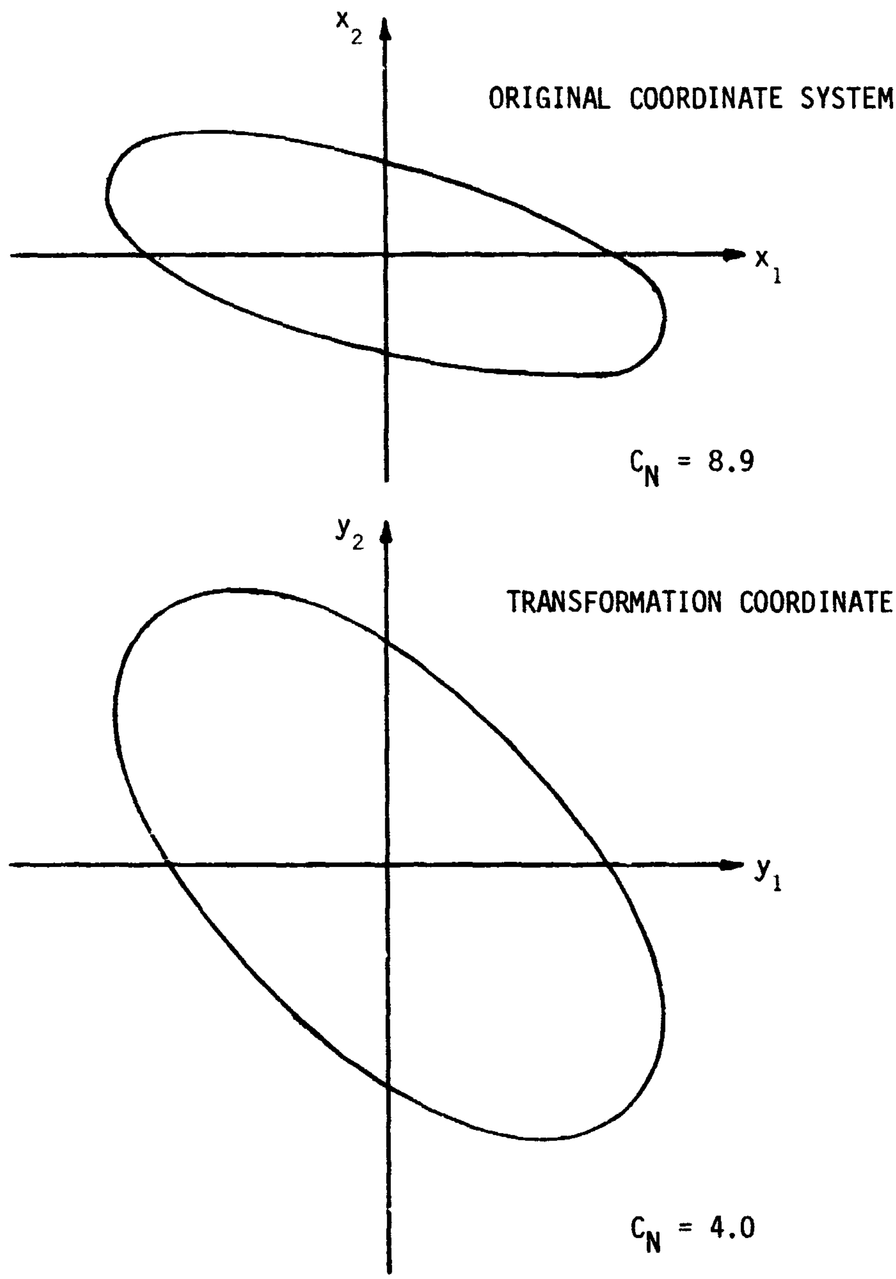

Figure 3 An Ellipsoid in its Original (x) and Transformed (y) Coordinate Systems

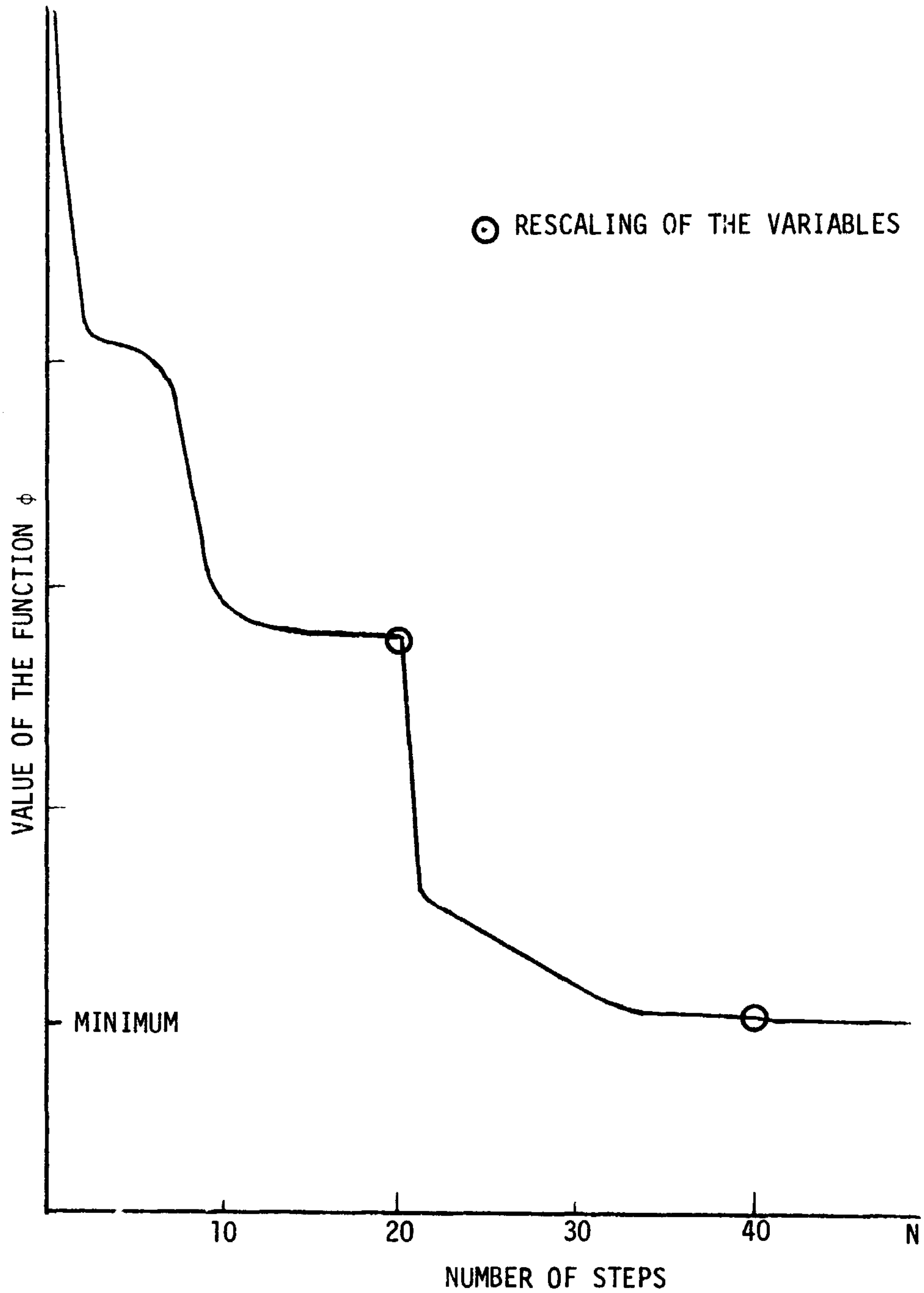

Figure 4 Value of ϕ as a Function of the Number of Steps -- Rescaling Every 20 Steps

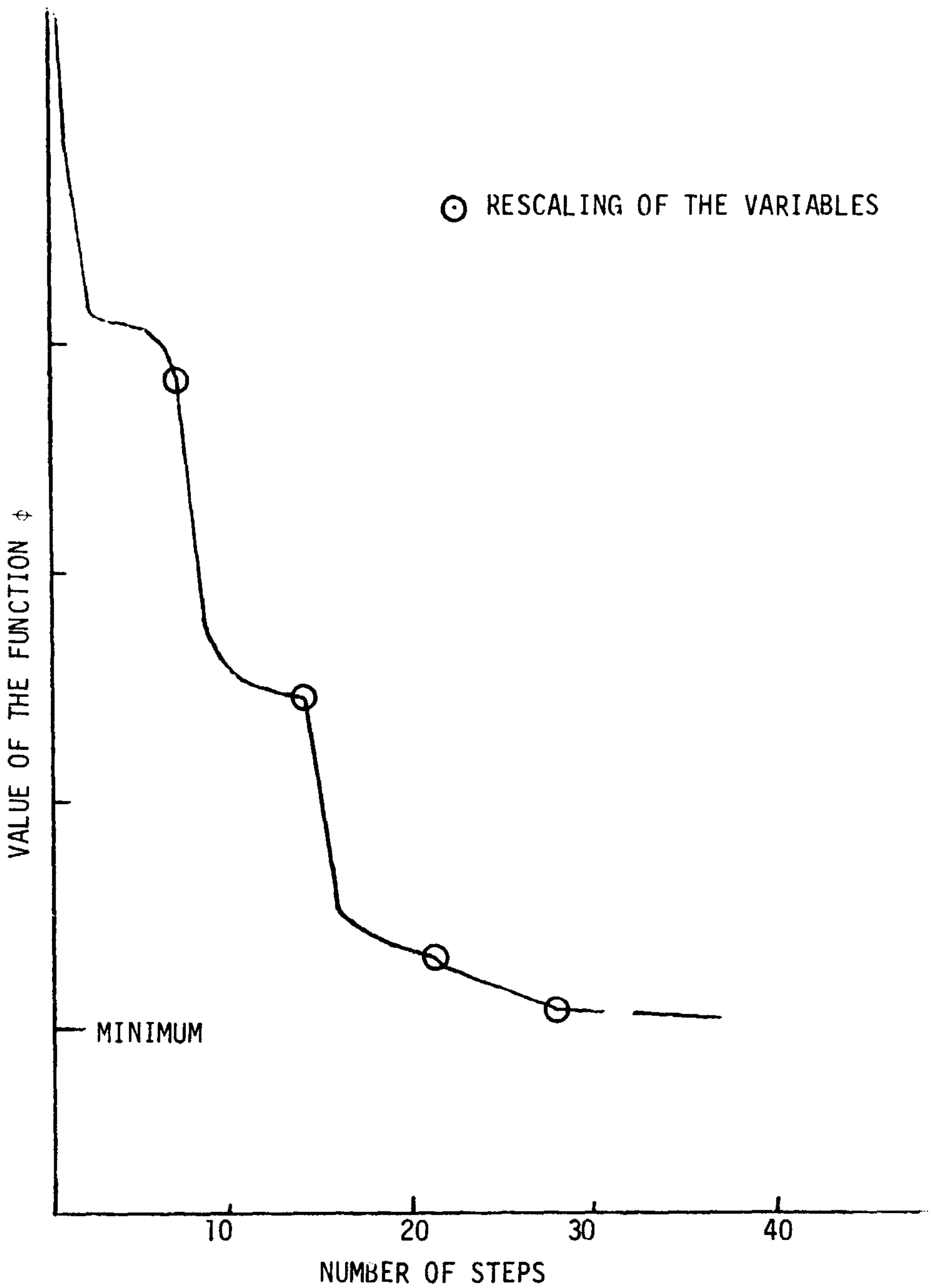

Figure 5 Value of ϕ as a Function of the Number of Steps -- Rescaling Every 7 Steps

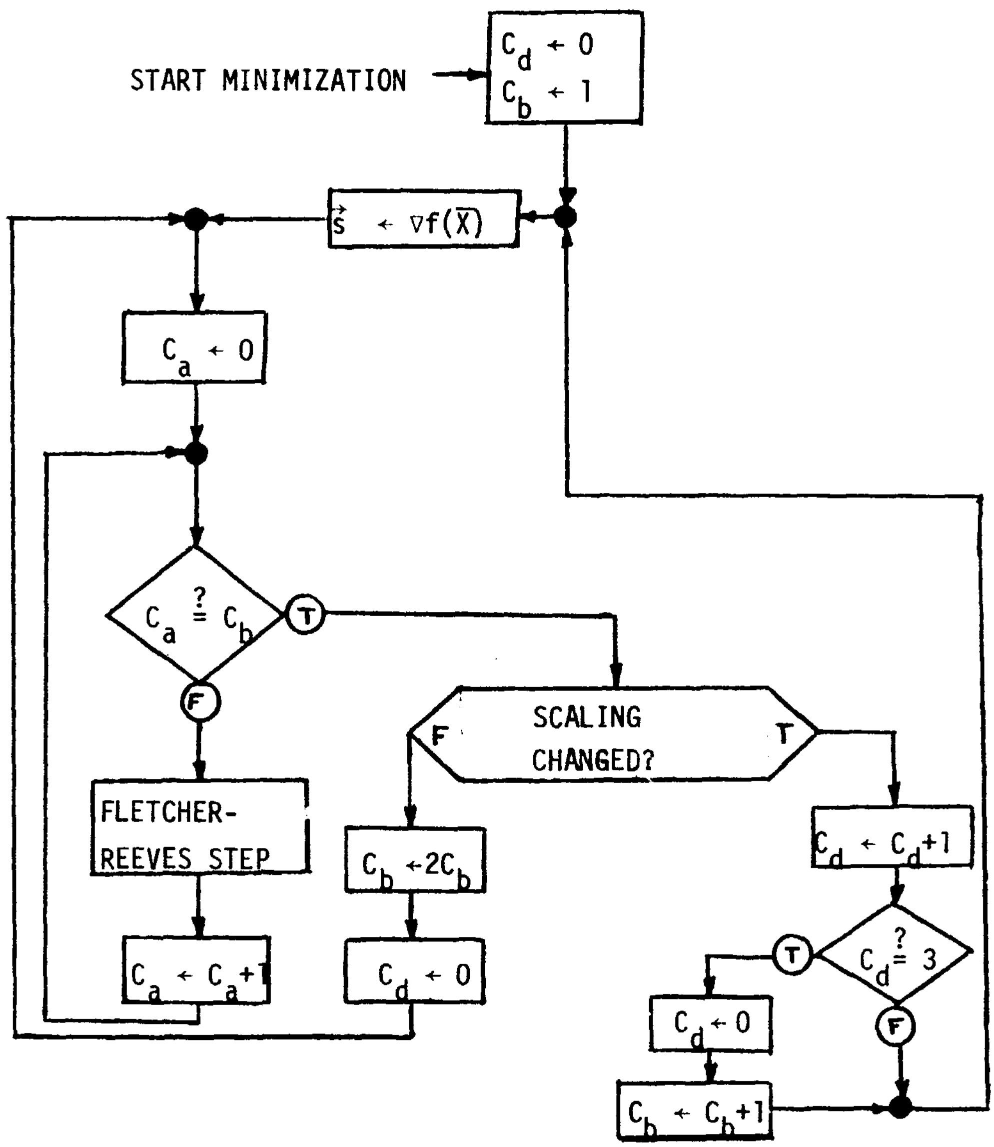

Figure 6 Flow Diagram for Frequency of Rescaling Logic

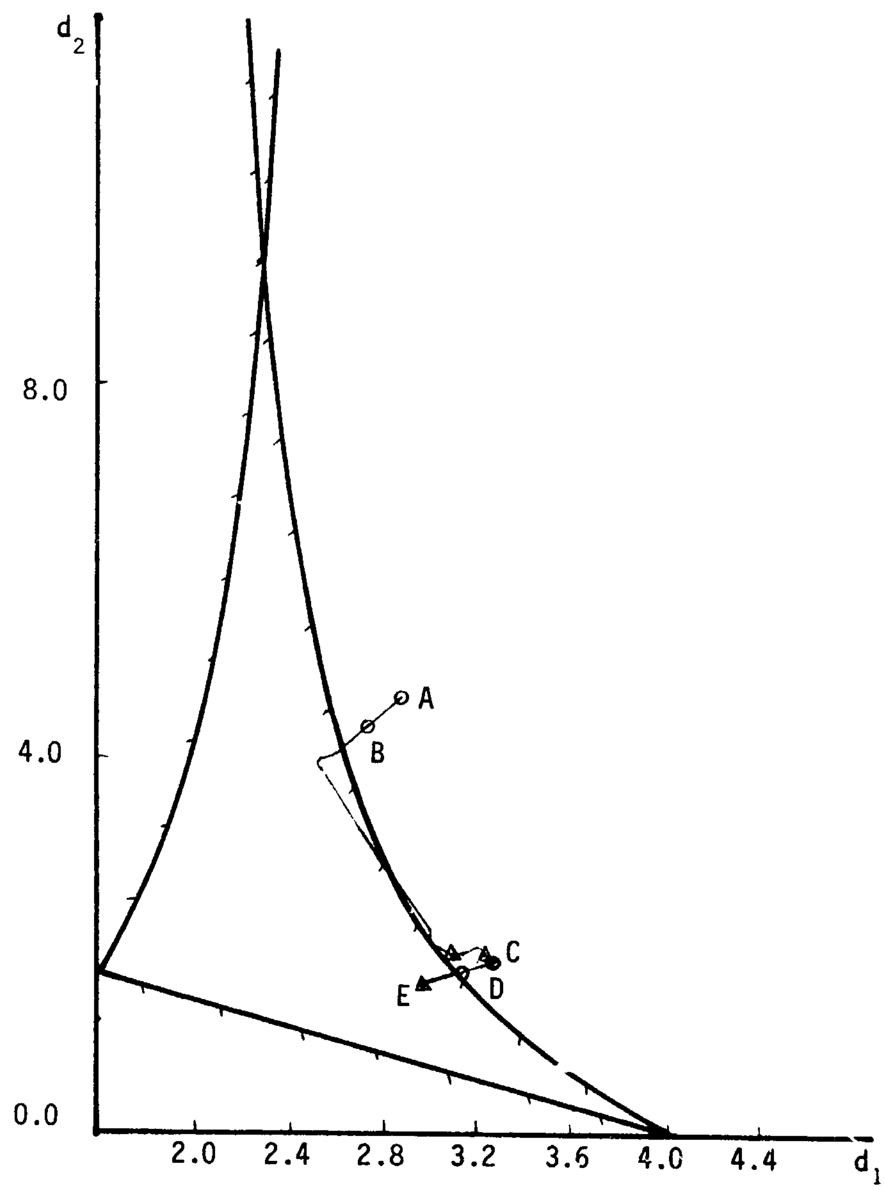

Figure 7 Synthesis Path for Three Bar Truss, Case 1

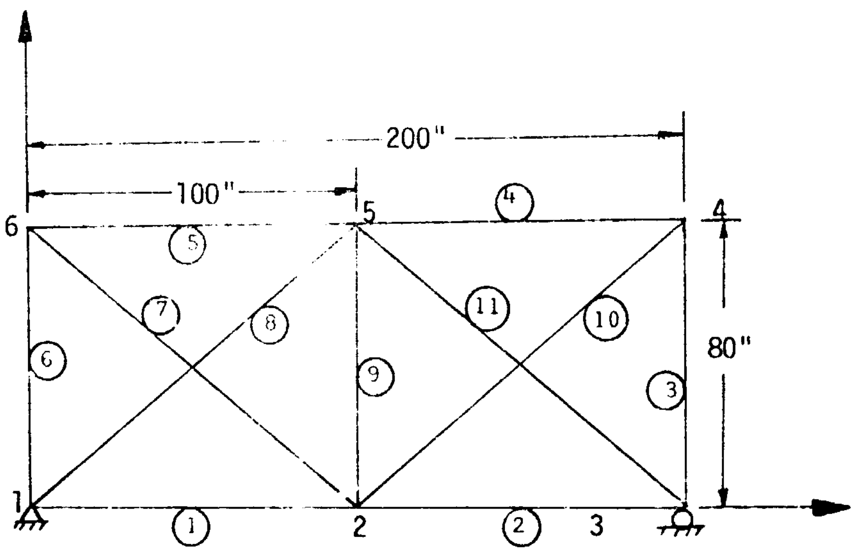

Figure 8 Eleven Bar Truss,
Case 2

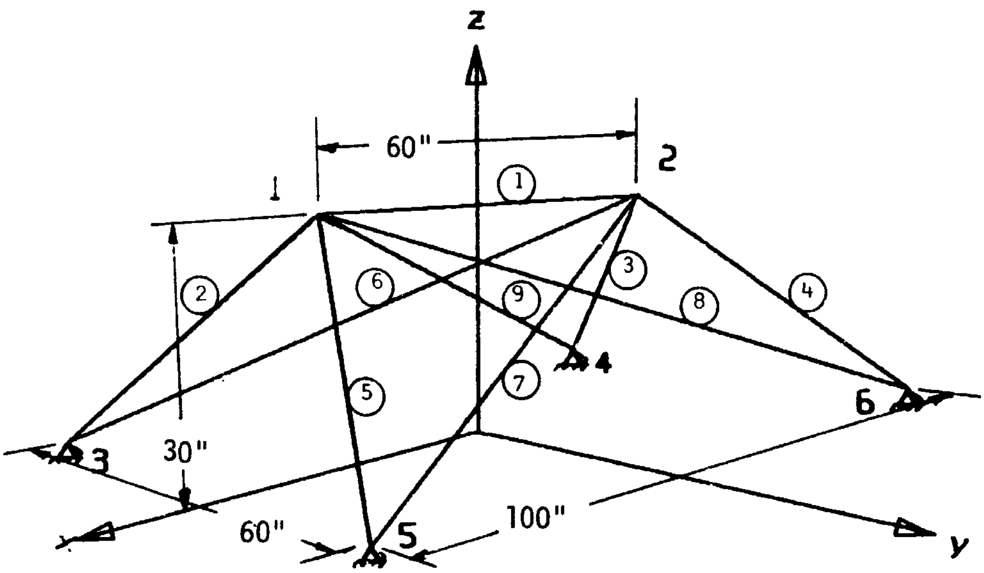

Figure 9 Nine Bar Truss,
Case 3

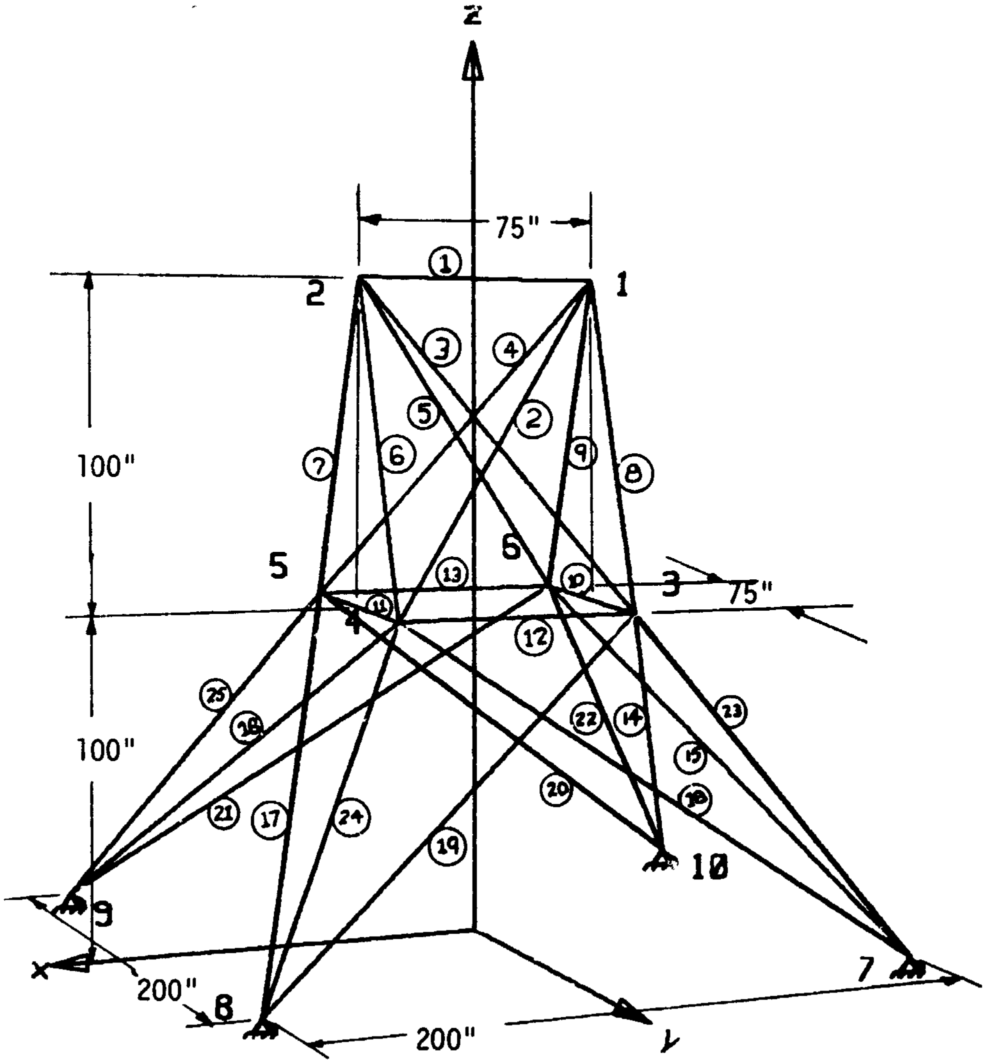

Figure 10 Twenty Five Bar Truss,
Case 4

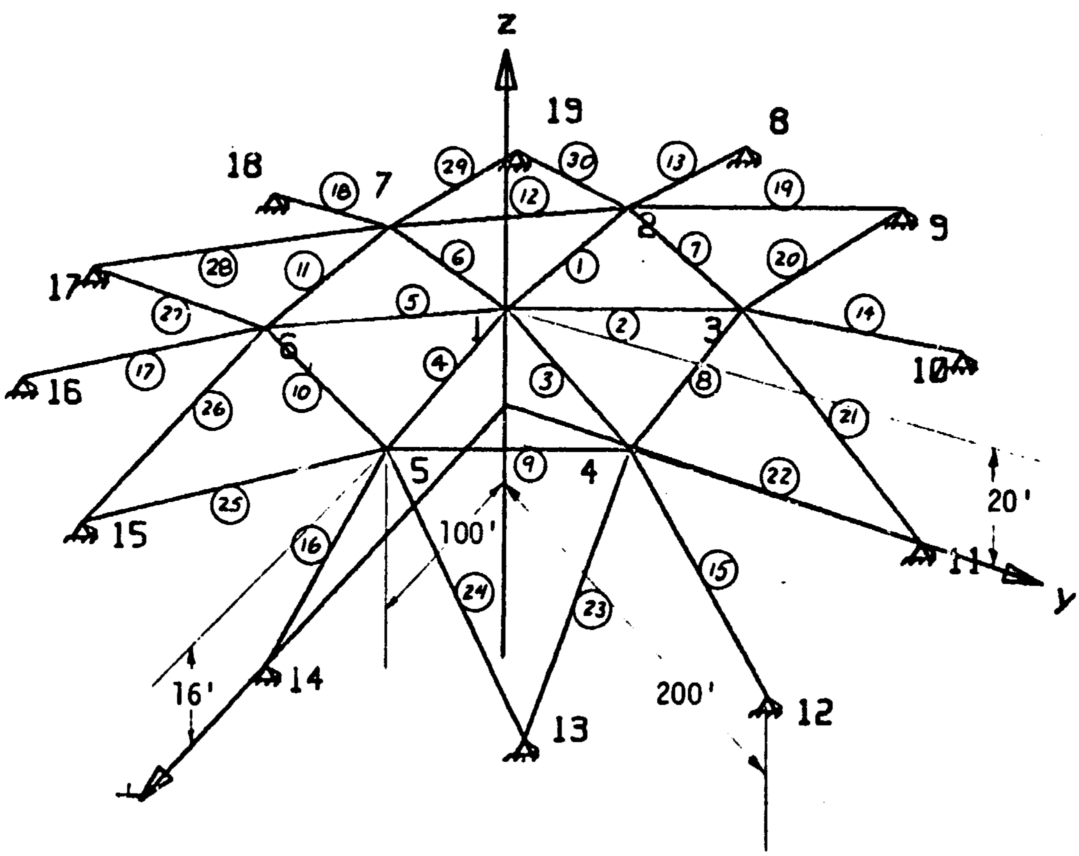

Figure 11 Thirty Bar Truss,
Case 5

TABLE 1 EIGENVALUES FOR MATRIX OF SECOND PARTIAL DERIVATIVES FOR THREE BAR TRUSS

	Arbitrary Point		Minimum	
	Unscaled	Scaled	Unscaled	Scaled
Largest Eigenvalue λ_{max}	9.95×10^5	2.33	1.35×10^6	2.39
Smallest Eigenvalue λ_{min}	2.22	0.114	1.54×10^{-1}	6.46×10^{-3}
Conditioning Number C_N	4.46×10^5	20.4	8.74×10^6	370

TABLE 2

SUMMARY OF RUNNING TIMES

Case	Time to Find Final Design (Sec)	Termination Time (Sec)
1	38	30
2A	10	417
2B	505	1728
3	555	302
4A	1595	694
4B	2216	1230
4C	1973	1159
5	1040	2221

TABLE 3, CASE 1

DISPLACEMENT LIMITS = $\pm$1.0 IN.

Member	Initial Diameters (in)	Final Diameters (in)	Optimum Design
1	2.88	3.148	3.15
2	4.64	1.729	1.64
3	2.88	3.148	3.15
Weight,lb	95.89	79.75	79.2

TABLE 4

LOAD CONDITIONS FOR CASE 2

Load Condition 1
(lbs)

$\vec{P}_{21}$	$\vec{P}_{41}$	$\vec{P}_{51}$	$\vec{P}_{61}$
-5000	-1000	-3000	1000
0	0	0	0

Load Condition 2
(lbs)

$\vec{P}_{22}$	$\vec{P}_{42}$	$\vec{P}_{52}$	$\vec{P}_{62}$
0	0	0	0
0	3000	0	0

Load Condition 3
(lbs)

$\vec{P}_{23}$	$\vec{P}_{43}$	$\vec{P}_{53}$	$\vec{P}_{63}$
0	0	0	0
0	0	0	-3000

TABLE 5, CASE 2.A

DISPLACEMENT LIMITS = ± 0.15 IN.

Member	Initial Diameters (in)	Final Diameters (in)	Results From Ref. 1 (in)
1	4.00	3.095	3.87
2	3.10	2.399	2.44
3	3.49	2.700	2.50
4	3.42	2.646	2.50
5	3.27	2.530	2.22
6	3.14	2.430	2.00
7	2.96	2.290	2.92
8	3.61	2.793	3.28
9	2.22	1.702	1.26
10	3.55	2.747	3.00
11	3.20	2.476	2.65
Weight,lb	106.0	82.06	82.6

TABLE 6, CASE 2.B

DISPLACEMENT LIMITS = ±0.25 IN.

Member	Initial Diameters (in)	Final Diameters (in)	Results From Ref. 1 (in)
1	4.00	1.740	1.85
2	3.10	1.425	1.32
3	3.49	1.719	1.69
4	3.42	1.960	1.84
5	3.27	1.678	1.69
6	3.14	1.594	1.61
7	2.96	.889	1.08
8	3.61	2.950	2.66
9	2.22	1.125	1.01
10	3.55	1.521	1.49
11	3.20	2.552	2.50
Weight,lb	106.0	57.3	56.1

TABLE 7

LOAD CONDITIONS FOR CASE 3

Load Condition 1 (lbs)		Load Condition 2 (lbs)	
$\vec{P}_{11}$	$\vec{P}_{21}$	$\vec{P}_{12}$	$\vec{P}_{22}$
0	0	0	4000
1700	-1700	0	0
0	0	0	-3000

TABLE 8, CASE 3

DISPLACEMENT LIMITS = ± 10.0 IN

Member	Initial Design		Final Design		Results From Reference 1	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	2.80	0.010	2.058	0.0109	2.16	.012
2,3,4,5	1.10	0.079	1.929	0.0133	1.80	.018
6,7,8,9	3.33	0.030	3.535	0.0180	3.55	.016
Weight, lb	11.062		5.971		6.03	

TABLE 9

LOAD CONDITIONS FOR CASE 4

Load Condition 1

(lbs)

$\vec{P}_{11}$	$\vec{P}_{21}$	$\vec{P}_{31}$	$\vec{P}_{61}$
1000	0	500	500
10000	10000	0	0
-5000	-5000	0	0

Load Condition 2

(lbs)

$\vec{P}_{12}$	$\vec{P}_{22}$	$\vec{P}_{32}$	$\vec{P}_{62}$
0	0	0	0
20000	-20000	0	0
-5000	-5000	0	0

TABLE 10, CASE 4.A

DISPLACEMENT LIMITS = ± 0.35 IN

Member	Initial Design		Final Design		Results From Reference 8	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	3.96	0.20	0.710	0.0243	2.74	0.091
2,3,4,5	3.96	0.20	3.270	0.2113	3.18	0.24
6,7,8,9	3.96	0.20	3.449	0.2149	2.80	0.25
10,11	3.96	0.20	1.011	0.0182	1.72	0.017
12,13	3.96	0.20	1.145	0.0685	3.35	0.034
14,15,16,17	3.96	0.20	3.603	0.0697	3.21	0.091
18,19,20,21	3.96	0.20	3.881	0.1655	3.97	0.15
22,23,24,25	3.96	0.20	3.333	0.2535	2.91	0.28
Weight, lb	822.9		562.9		570.4	

TABLE 11, CASE 4.B

DISPLACEMENT LIMIT = ± 10.0 IN.

UPPER d/t RATIO LIMIT = 100.0

Member	Initial Design		Final Design	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	.80	0.025	0.920	0.0393
2,3,4,5	3.30	0.220	4.189	0.0873
6,7,8,9	3.45	0.220	3.567	0.1346
10,11	1.05	0.020	0.691	0.0152
12,13	1.15	0.070	1.879	0.0216
14,15,16,17	3.65	0.070	4.242	0.0433
18,19,20,21	3.90	0.170	5.608	0.0574
22,23,24,25	3.35	0.260	5.020	0.0509
Weight, lb	581.3		285.6	

TABLE 12, CASE 4.C

DISPLACEMENT LIMIT = ± 10.0 IN

UPPER d/t RATIO LIMIT = 400.0

Member	Initial Design		Final Design	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1	0.920	0.0393	0.852	0.0245
2,3,4,5	4.184	0.0873	5.164	0.0442
6,7,8,9	3.567	0.1346	4.055	0.0890
10,11	0.691	0.0152	0.699	0.0093
12,13	1.879	0.0216	1.700	0.0083
14,15,16,17	4.242	0.0433	5.275	0.0220
18,19,20,21	5.608	0.0574	6.690	0.0338
22,23,24,25	5.020	0.0509	5.472	0.0405
Weight, lb	285.6		202.3	

TABLE 13

LOAD CONDITION FOR CASE 5

Load Condition 1 (ksi)

$\vec{P}_{11}$	$\vec{P}_{21}$	$\vec{P}_{31}$	$\vec{P}_{41}$	$\vec{P}_{51}$	$\vec{P}_{61}$	$\vec{P}_{71}$
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
-78.5	125.0	125.0	125.0	125.0	125.0	125.0

Load Condition 2 (ksi)

$\vec{P}_{12}$	$\vec{P}_{22}$	$\vec{P}_{32}$	$\vec{P}_{42}$	$\vec{P}_{52}$	$\vec{P}_{62}$	$\vec{P}_{72}$
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
59.0	187.5	187.5	187.5	0.0	0.0	0.0

TABLE 14, CASE 5

DISPLACEMENT LIMITS = ± 1.60 IN.

Member	Initial Design		Final Design	
	Diameters (in)	Thicknesses (in)	Diameters (in)	Thicknesses (in)
1 → 6	15.0	1.0	12.25	0.817
7 → 12	15.0	1.0	13.31	0.888
13 → 18	15.0	1.0	12.30	0.820
18 → 30	15.0	1.0	14.16	0.844
Weight.lb	266600		199700	

TABLE 15

COMPARISON OF RUNNING TIMES

Case	For Results From Algorithm: To Achieve Final Design[a] (sec)	For Results From Algorithm: Termination Time[a] (sec)	From Previous Results (Ref. 1): To Achieve Final Design (sec)
2.A	10	417	221[b,c]
2.B	505	1728	1532[b]
3	555	302	3561[b]
4.A	1595	694	363[d]

a. Using the Fletcher-Reeves minimization method with scaling.

b. Using a modified steepest descent minimization scheme.

c. Estimate termination time was 375 seconds.

d. Using a variable metric minimization scheme.

TABLE 16

COMPARISON OF RESULTS

Case	Final Weights Using New Algorithm (lb)	Final Weights Using Comparable Methods[a] (lb)	Lowest Estimate of Optimum Weight (lb)
1	79.75	-----	79.2[d]
2.A	82.06	82.6[b]	79.6[b]
2.B	57.30	56.1[b]	55.4[b]
3	5.97	6.03[b]	5.96[b]
4.A	562.9	570.4[c]	562.9

a. Running times for these results given in Table 15.

b. References 1 or 8.

c. Reference 8.

d. Reference 13.

LIST OF REFERENCES

1. Fox, R. L., "An Integrated Approach to Engineering Synthesis and Analysis", Ph.D. Thesis, Engineering Division, Case Institute of Technology, Cleveland, Ohio, 1965.

2. Shanley, F. R., Weight-Strength Analysis of Aircraft Structures, (2nd Edition, Dover Publications, Inc., New York, 1960).

3. Gellatly, R. A., Gallagher, R. H., Luberacki, W. A., "Development of a Procedure for Automated Synthesis of Minimum Weight Structures", Air Force Contract AF 33(657)-11748, Preliminary Copy.

4. Schmit, L. A., and Mallett, R. H., "Structural Synthesis and Design Parameter Hierarchy," J. Struct. Div. Proc. Am. Soc. Civil Engrs. 89, No. ST4, 269-299, (August, 1963).

5. Micks, W. R., "Bibliography of Literature on Optimum Design of Structures and Related Topics", RM 2304, The Rand Corporation, Santa Monica, California, 1958.

6. Wasiutynski, Z. and Brandt, A., "The Present State of Knowledge in the Field of Optimum Design of Structures," Applied Mechanics Reviews, Vol. 16, No. 5, May 1963, pp. 341-350.

7. Gerard, G., "Optimum Structural Design Concepts for Aerospace Vehicles: Bibliography and Assessment", ARA Technical Report No. 272-2, March 1965.

8. Fox, R. L. and Schmit, L. A., "Advances in Integrated Approach to Structural Synthesis", J. of Spacecraft and Rockets, Vol. 3, No. 6, pp. 858-866, 1966.

9. Norris, C. H. and Wilbur, J. B., Elementary Structural Analysis, (2nd Edition, McGraw-Hill Book Company, New York, 1962).

10. Fox, R., and Stanton, E. L., "Developments in Structural Analysis by Direct Energy Minimization," Case Institute of Technology of Case Western Reserve University. To be published in the AIAA Journal, Summer, 1968.

11. Fletcher, R. and Reeves, C. M., "Function Minimization by Conjugate Gradients", British Computer J., Vol. 7, No. 2, pp. 149-154, 1964.

12. Fletcher, R. and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization," British Computer J., Vol. 6, No. 2, pp. 163-168, 1963.

13. Schmit, L. A., "Comment on 'Completely Automatic Weight Minimization Method for High-Speed Digital Computers'", J. of Aircraft, Vol. 1, No. 6, pp. 375-377, 1964.

APPENDIX A

FORMULAS

This appendix contains the formulas for the gradient and the diagonal elements of the matrix of second partial derivatives for the function ϕ presented in Chapter III.

A. Gradients

$$\frac{\partial\phi}{\partial d_i} = 2\sum_{k=1}^{KT}\sum_{\ell=1}^{N}\left[\left(\sum_{j=1}^{N} K_{\ell j}u_{jk} - P_{\ell k}\right)\sum_{p=1}^{N}\left(\frac{\partial K_{\ell p}}{\partial d_i}u_{pk}\right)\right]$$

$$+ \nu\left\{\frac{1}{(d_{max} - d_i)^2} - \frac{1}{d_i^2} + \frac{1}{t_i(r_u - d_i/t_i)^2}\right.$$

$$- \frac{1}{t_i(d_i/t_i - r_\ell)^2} + \sum_{k=1}^{KT}\left[- \frac{2K_d\pi^2 E_i d_i}{8L_i^2 C_{z1}\left(\sigma_{ik} + \frac{\pi^2 E_i(d_i^2+t_i^2)}{8L_i^2}\right)^2}\right.$$

$$\left. + \frac{K_d k_2 E_i t_i}{d_i^2\left(\sigma_{ik} + \frac{k_2 E_i t_i}{d_i}\right)^2}\right]$$

$$\left. + \frac{\pi\rho W_m W_o L_i t_i}{(W_o - W)^2}\right\}$$

$$\frac{\partial\phi}{\partial t_i} = 2\sum_{k=1}^{KT}\sum_{\ell=1}^{N}\left[\left(\sum_{j=1}^{N} K_{\ell j}u_{jk} - P_{\ell k}\right)\sum_{p=1}^{N}\left(\frac{\partial K_{\ell p}}{\partial t_i}u_{pk}\right)\right]$$

$$+ \nu\left\{\frac{1}{(t_{max} - t_i)^2} + \frac{d_i}{t_i^2\left(\frac{d_i}{t_i} - r_\ell\right)^2}\right.$$

$$- \frac{d_i}{t_i^2 \left(r_u - \frac{d_i}{t_i} \right)^2}$$

$$+ \sum_{k=1}^{KT} \left[- \frac{2K_d \pi^2 E_i t_i}{8L_i^2 C_{z1} \left(\sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2} \right.$$

$$\left. - \frac{K_d k_2 E_i}{d_i \left(\sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^2} \right]$$

$$+ \frac{\pi \rho W_m W_o L_i d_i}{(W_o - W)^2} \Bigg\}$$

$$\frac{\partial \phi}{\partial u_{jk}} = 2 \sum_{i=1}^{N} K_{ji} \left(\sum_{\ell=1}^{N} K_{i\ell} u_{\ell k} - P_{ik} \right)$$

$$+ \nu \left\{ \sum_{i=1}^{M} \sum_{v=1}^{6} n \left(\frac{j}{SR_{iv}} \right) \left(\frac{K_d E_i \gamma_{ij}}{L_i} \right) \right. \Bigg[$$

$$\frac{1}{(\sigma_{yt} - \sigma_{ik})^2} - \frac{1}{(\sigma_{ik} - \sigma_{yc})^2}$$

$$- \frac{1}{C_{z1} \left(\sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2}$$

$$- \frac{1}{\left(\sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^2} \Bigg]$$

$$+ \frac{K_{uz}}{(u_{up} - u_{jk})^2} - \frac{K_{uz}}{(u_{jk} - u_{\ell o})^2} \Bigg\}$$

B. Second Partial Derivatives

$$\frac{\partial^2 \phi}{\partial d_i^2} = 2 \sum_{k=1}^{KT} \sum_{\ell=1}^{N} \Big(\sum_{j=1}^{N} u_{\ell j} \frac{\partial K_{\ell j}}{\partial d_i} \Big)^2$$

$$+ \nu \Bigg\{ \frac{2}{(d_{max} - d_i)^3} + \frac{2}{d_i^3}$$

$$+ \frac{2}{t_i^2(r_u - d_i/t_i)^3} + \frac{2}{t_i^2(d_i/t_i - r_\ell)^3}$$

$$+ \sum_{k=1}^{KT} \Bigg[\frac{2K_d}{C_{z1}\left(\sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2}\right)^3} \left(\frac{2\pi^2 E_i d_i}{8L_i^2}\right)^2$$

$$- \frac{2K_d \pi^2 E_i}{8L_i^2 C_{z1}\left(\sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2}\right)^2}$$

$$+ \frac{2K_d}{\left(\sigma_{ik} + \frac{k_2 E_i t_i}{d_i}\right)^3} \left(\frac{k_2 E_i t_i}{d_i^2}\right)^2$$

$$- \frac{2K_d k_2 E_i t_i}{d_i^3 \left(\sigma_{ik} + \frac{k_2 E_i t_i}{d_i}\right)^2} \Bigg]$$

$$+ \frac{2\pi^2\rho^2 W_m W_o L_i^2 t_i^2}{(W_o - W)^3} \Bigg\}$$

$$\frac{\partial^2 \phi}{\partial t_i^2} = 2 \sum_{k=1}^{KT} \sum_{\ell=1}^{N} \left(\sum_{j=1}^{N} u_{\ell j} \frac{\partial K_{\ell j}}{\partial t_i} \right)^2$$

$$+ \nu \Bigg\{ \frac{2}{(t_{max} - t_i)^3} + \frac{2}{(d_i/t_i - r_\ell)^3} \left(\frac{d_i}{t_i^2} \right)^2$$

$$- \frac{2d_i}{t_i^3 (d_i/t_i - r_\ell)^2} + \frac{2}{(r_u - d_i/t_i)^3} \left(\frac{d_i}{t_i^2} \right)^2$$

$$+ \frac{2d_i}{t_i^3 (r_u - d_i/t_i)^2}$$

$$+ \sum_{k=1}^{KT} \Bigg[\frac{2K_d}{C_{z1} \left(\sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^3} \left(\frac{2\pi^2 E_i t_i}{8L_i^2} \right)^2$$

$$- \frac{2K_d \pi^2 E_i}{8L_i^2 C_{z1} \left(\sigma_{ik} + \frac{\pi^2 E_i (d_i^2 + t_i^2)}{8L_i^2} \right)^2}$$

$$+ \frac{2K_d k_2^2 E_i^2}{d_i^2 \left(\sigma_{ik} + \frac{k_2 E_i t_i}{d_i} \right)^3} \Bigg]$$

$$+ \frac{2\pi^2\rho^2 W_m W_o L_i^2 d_i^2}{(W_o - W)^3} \Bigg\}$$

$$\frac{\partial^2\phi}{\partial u_{jk}^2} = 2K_{jj}^2 + \nu\left\{\sum_{i=1}^{M}\sum_{v=1}^{b}\eta\left(\frac{j}{SR_{iv}}\right)\left(\frac{K_d E_i^2 \gamma_{iv}^2}{L_i^2}\right)\left[\frac{2}{(\sigma_{max}-\sigma_{ik})^3} + \frac{2}{(\sigma_{ik}-\sigma_{min})^3}\right.\right.$$

$$+ \frac{2}{C_{z1}\left(\sigma_{ik} + \dfrac{\pi^2 E_i(d_i^2+t_i^2)}{8L_i^2}\right)^3}$$

$$\left. + \frac{2}{\left(\sigma_{ik} + \dfrac{k_2 E_i t_i}{d_i}\right)^3}\right]$$

$$\left. + \frac{2K_{uz}}{(u_{up}-u_{jk})^3} + \frac{2K_{uz}}{(u_{jk}-u_{\ell o})^3}\right\}$$

C. The partial derivatives of the design and displacement variables were coordinated to the thickness, diameter, and displacements in the following manner:

<u>Displacements</u>

$$\frac{\partial\phi}{\partial X_i} = \frac{\partial\phi}{\partial u_{jk}} \qquad\qquad \frac{\partial^2\phi}{\partial X_i^2} = \frac{\partial^2\phi}{\partial u_{jk}^2}$$

$$i = j + nk$$

$$j = 1,2,\ldots,n$$

$$k = 1,2,\ldots,KT$$

Diameters

$$\frac{\partial \phi}{\partial \alpha_{i-nKT}} = \frac{\partial \phi}{\partial X_i} = \sum_{j=1}^{M} \eta \left(\frac{DD_j}{i-nKT} \right) \frac{\partial \phi}{\partial d_i}$$

$$\frac{\partial^2 \phi}{\partial \beta^2_{i-nKT}} = \frac{\partial^2 \phi}{\partial X_i^2} = \sum_{j=1}^{M} \eta \left(\frac{DD_j}{i-nKT} \right) \frac{\partial^2 \phi}{\partial d_i^2}$$

$$i = nKT + 1, \ldots, nKT + N_{dv}$$

Thicknesses

$$\frac{\partial \phi}{\partial \beta_{i-nKT-N_{dv}}} = \frac{\partial \phi}{\partial X_i} = \sum_{j=1}^{M} \eta \left(\frac{TT_j}{i-nKT-N_{dv}} \right) \frac{\partial \phi}{\partial t_i}$$

$$\frac{\partial \phi}{\partial \beta^2_{i-nKT-N_{dv}}} = \frac{\partial^2 \phi}{\partial X_i^2} = \sum_{j=1}^{M} \eta \left(\frac{TT_j}{i-nKT-N_{dv}} \right) \frac{\partial^2 \phi}{\partial t_i^2}$$

$$i = nKT + N_{dv}+1, \ldots, nKT + N_{dv} + N_{tv}$$

APPENDIX B

The following are the listings of the ALGOL programs used to obtain the results presented in Chapter 5:

1. an ALGOL procedure, FP, to compute the value of the function ϕ, its gradient, and the scaling factors;
2. the main program used to minimize the function ϕ using the Fletcher-Reeves method.

```
A ALG   FUNPHI
COMPILED BY CASE 1107 ALGOL 60 (FAST VERSION)  DATED  DECEMBER 14, 1967
THIS COMPILATION WAS DONE ON 08 APR 68 AT 10:11:57
B0                      PROCEDURE FP(ROW,X,DELX,PHEE,B1,B2,B3,   CHEK)$
BLOCK 1     LEVEL 1
BLOCK 2     LEVEL 2
                              INTEGER ROW,CHEK $
                              REAL PHEE $
                              REAL ARRAY X,DELX $
                              BOOLEAN B1,B2,B3     $
                        BEGIN
B1                            OWN INTEGER KT,N,M,NN, NTV, NDV $
                              IF B1 THEN READ (KT,N,M, NN, NDV , NTV )$
                        BEGIN
B2                      OWN REAL ARRAY KLOCAL(1:M,1:3,1:3),EGL2, DRC(1:M,1:3), BUC,SMIN,SMAX,
BLOCK 3     LEVEL 3
                                  KCONST, E, D, I, EAL, AE, LEN(1:M), P , U , RVS(0:N,1:KT),
                                  ALPHA(0:NDV) , BETA(0:NTV), COORD(1:NN,1:3),
                                  TSCALE(1:N*KT + NDV + NTV  )$
                        REAL ARRAY LITTLE(0:N),STRESS(1:M,-1:KT), F(1:6,1:2), Y(1:2,1:6),
                                  LL(1:2,1:2), XXD(1:3),PARTU,PUR,PLOC,LUCU,PLC,KLC(1:6),
                                  RV(0:N)                 $
                        REAL ARRAY TSKALE (1:N*KT + NDV + NTV )$
                        OWN REAL  RNU5,G1I,G2I,G3I,G4I,C1,C2,C3,C4 ,C21I $
                        OWN REAL RAL,RAU,DMAX,TMAX,.JZ$
                        OWN REAL C21,C22,DIR.,MJ, NU, P1, P2, P3, PI, RHO, RNU, SAVPHI, SFC,
                                  SFE, THG , TS, TS1, W, WM, WO , ULO, UUP , AAA,BBB   $
                        OWN INTEGER ARRAY SR(1:M,1:6) , PT(1:M,1:2), UU(1:3*NN),SRSM(1:6),
                                  DD,TT(1:M)                $
                        REAL  AA , ALPH, BET, BB, CON1, CON2,  CON3, CON4, CON5, CON6, CON7,
                                  CON8, CON9, G1, G2, G3, G4, G5, G6, G15, G25, G35, G45,
                                  G1C, G2C, G3C, G4C, MESS , R , SIG , TTT           $
                        OWN REAL WA,WB,MXE,NU1,NU2,INC       $
                        INTEGER ALT, B, C, CHECK,   I, ID, IT, J, K, KN, L, NO, PT1, PT2, ROM,
                                  U1, U2, V          $
                        OWN INTEGER NKT, NKTO, NKTOT, NUDIV    ,NK     $
                        OWN INTEGER   CTJ $
                        OWN BOOLEAN B4, B5, B6, B7, B8, B9, B10, RESIDUALS$
                        LIST LISTA( J,E(J), PT(J,1) , PT(J,2)   ,D(J), DD(J) , T(J) , TT(J),
                                      SMAX(J) , SMIN(J)       )$
                        LIST LISTB( J, COORD(J,1) , COORD(J,2) , COORD(J,3), V, B, L)$
                        FORMAT F1A(8R16.8,A1)$
@                       FORMAT FORA( X20, 'MINIMUM', A1.1, X5, 'MEMBER', X8, 'DIAMETER',
                             X9, 'WALL THICKNESS', X10,'RATIO D(I)/T(I)', A1.1  )$
@                       FORMAT FORB( X6, 'VALUE OF RESIDUALS', A1.1, X38, 'LOADING', A1.1,
                             ' NODE    COMP',A0.1, X19, 7( I2, X13), A1.1 )$
@                       FORMAT FORC( 'NODAL DISPALCEMENTS', A5.1, X38,'LOADING', A1.1,
                             ' NODE    COMP',A0.1 , X19, 7(I2,X13), A1.1 )$
@                       FORMAT FORD( 'NEW NU = ',R16.8, A1.1, 'NEW WM = ', R16.8, A1.1,
                             'NEW WEIGHT CONSTRAINT', R16.8, A1.1 )$
                        FORMAT FORE('VALUE OF MJ = ',D10.2,A1)$
                        FORMAT EJ( E0,A1)$
                        FORMAT FR1( X6,I3,X9,D9.5,X11 , D11.7, X14, D8.2, A1.1)$
```

```
FORMAT FR2('    ',I5,A1.0)$
FORMAT FR2X( X11,'X ', (D12.1,X3),A0.1        )$
FORMAT FR2Y( X11,'Y ',7(D12.1,X3),A0.1        )$
FORMAT FR2Z( X11,'Z  ',7(D12.1,X3),A0.4       )$
FORMAT FR3( X5, 'NEW NU IS NOW SET EQUAL TO ',X3,R16.8,A3.1)$
FORMAT FR4X( X11,'X ',   7(D12.9  , X3), A0.1)$
FORMAT FR4Y( X11,'Y ',   7(D12.9  , X3), A0.1)$
FORMAT FR4Z( X11,'Z ',   7(D12.9  , X3), A0.4)$
FORMAT FR5('      MEMBER', X15,5I15, A5.1)$
FORMAT FR6('  YIELD IN TENSION', X9, 5D15.1,A1.0)$
FORMAT FR7('  YIELD IN COMPRESSION', X5, 5D15.1, A1.0)$
FORMAT FR8('  EULER STRESS', X13, 5D15.1,A1.0)$
FORMAT FR9('  CRIPPLING STRESS', X9, 5D15.1, A1.1)$
FORMAT FR10( I7, X2?, 5D15.1, A1.0)$
FORMAT FR11( E1,A1.0,X25,'INPUT DATA',A1.2,
             X2,'MEMBER INFORMATION: ',A0.2,
             X43,'CONTROLING',A1,
             X29,'CONNECTS',X3,'DIAM     ',X2,'THICKNESS',X8,'INITIAL'
                 ,X8 'INITIAL',
                  X?,'YIELD STRESSES',A1,
             '  MEMBER',X9,'E(I)',X9,'PT1  PT2',X3,'VARIABLE',X3,
                 'VARIABLE',
                  X7,'DIAMETER',X7,'THICKNESS',X7,'UPPER',
                  X7,'LOWER',A1.1)$
FORMAT FR12(I6,X7,R11.4,X5,2I4,I7,X6,I4,X10,D8.4,X7,D9.6,X2,
             2R12.3,A1)$
FORMAT FR13( E1,A1,
             X10,'CONSTANTS USED IN PROBLEM',A0.1,
             '  UPPER AND LOWER DISPLACEMENT LIMITS',X10,'UUP =',
                  D8.3,' IN',X5,'ULO =',D8.3,' IN',A1.1,
             '  MAXIMUM DIAMETER AND THICKNESS',X15,'DMAX=',D5.1,
                  X3,'IN',X8,'TMAX=',D5.5,A1.1,
             '  UPPER AND LOWER  D(I)/T(I)  RATIOS',X11,'RAU =',X1,
                  D7.1,X11,'RAL =',D7.1,A1.1,
             '  SAFETY FACTORS',X31,'SFE =',D5.1,X13,'SFC =',D5.1,A1.1,
             '  MXE =',D10.1,A1.1,
             '  WO =',D14.4,A1.1,
             '  RHO = ',D9.4,'  LB/CU IN',A1.1)$
FORMAT FR14( E1,A1,
             X5, 'NODAL INFORMATION:',A2.2,
             X30,'COORDINATES',A1.1,
             X5, 'NODE',X12,'X',X14,'Y',X14,'Z',A1)$
FORMAT FR15( I6,X9,D7.1,X8,D7.1,X8,D7.1,A1)$
FORMAT FR16( E1,A1,
             X5,'LOADINGS',A1.2)$
FORMAT FR17( 'LOAD = ',I6,A3.2,
             X5, 'NODE',X11,'X',X14,'Y',X14,'Z',A1)$
FORMAT FR18( I6,X7,D8.1,X7,D8.1,X7,D8.1,A1)$
LIST LS1( FOR I = (1,1,M) DO  (I,ALPHA( DD(I) )*D(I),
      BETA( TT(I) )*T(I), ALPHA( DD(I) )*D(I)/( BETA( TT(I))*T(I) )))   $
LIST LS2( FOR I = (1,1,KT) DO I)$
LIST LS3( FOR K = (1,1,KT) DO RVS( UU(3*I-3+J),K))$
LIST LS4( FOR K = (1,1,KT) DO   S( UU(3*I-3+J),K))$
LIST LS5(I,E(I),PT(I,1),PT(I,2),DD(I),TT(I),ALPHA(DD(I))*D(I),
```

```
                                     BETA( TT(I))*T(I),SMAX(I),SMIN(I).*
                                 LIST LS6(UUP,ULO,DMAX,TMAX,RAU,RAL,SFE,SFC,SYF,WO,RHO)$
                                 LIST LS7( I,FOR J = 1,2,3 DO COORD(I J))$
                                 LIST LS8(I,FOR J = 1,2,3 DO P(UU(3*I-3+J),K) )$
                                 COMMENT THE VALUE OF THE RESIDUALS FOR NU EQUAL TO ZERO $
                                 PROCEDURE EXTRAP (TFA,NU,R,TFB)$
BLOCK 4      LEVEL 4
                                     BOOLEAN TFA,TFB$
                                     REAL NU,R$
                                 BEGIN
B3                               REAL SP $
                                 OWN REAL R1,R2,R3,N1,N2,N3,T1,T2,T3,S1,S2,A,C,B,DET      $
                                 IF NOT TFA THEN BEGIN
B4                                   R1 = R2 = R3 = N1 = N2 = N3 = S1 = S2 = A= B= C= 0.0 $
FR      E4                           GO TO DONE $    END$
                                 R3 = R2 $ R2 = R1 $ R1 = R $
                                 N3 = N2 $ N2 = N1 $ N1 = SQRT (NU)$
FR                               IF R2 EQL 0.0 THEN GO TO DONE $
                                 S2 = S1 $ S1 = (R2 - R1)/(N2 - N1)$
                                 C = B $   B = A $ A = R1 - S1*N1 $
                                 IF R3 NEQ 0.0 THEN BEGIN
B5                                   T2 = 1.0$ T3 = SQRT( NU0IV)$    T1 = 1.0/T3 $
                                     DET = (T3*T3 - T2*T2)*T1 + (T1*T1 - T3*T3)*T2
                                           + (T2*T2 - T1*T1)*T3 $
                                     AA = (  T1*T1*(T2*R3 - T3*R2) +  T3*T3*(T1*R2 - T2*R1)
                                        + T2*T2*(T3*R1 - T1*R3)    ) / DET $
                                     WRITE (AA)$
E5                                 END$
                                 SP = (-7.0*S2 + 29.0*S1)/24.0 $
                                 IF B NEQ 0.0 THEN
                                     IF B GTR CZ2 AND
                                        MIN ( AA,R1 - SP*N1) GTR CZ2
                                           THEN TFB = TRUE $
                                 WRITE (A,B,R,NU,CZ2,SP,S1)$
                                 DONE..
E3                                                                     END$
END BLOCK 4    LEVEL 4
                                 REAL PROCEDURE SOS(G,J)$
BLOCK 5      LEVEL 4
                                     REAL G$ INTEGER J $ BEGIN
B6                                   IF G LSS 10.0**J THEN SOS = 0.0 ELSE SOS =  G $
E6                                   END$
END BLOCK 5    LEVEL 4
                                 COMMENT MODIFY THE DESIGN SO THAT IT WILL SATISFY A REDUCED GOAL WEIGHT$
                                 PROCEDURE MODIF( U,ALPHA,BETA,WO,INC, M,ESTEP)$
BLOCK 6      LEVEL 4
                                     REAL ARRAY U,ALPHA,BETA$    REAL WO,INC,ESTEP$
                                     INTEGER M $
                                 BEGIN
B7                                   INTEGER I,J,K,II,ID    $
                                     INTEGER JJ $
                                     REAL A,B,DS,TS,W,BS,CS,AA,ALPH,BET $
                                     BOOLEAN B1 , B2 $
                                     INTEGER ARRAY JD,JT(1:M)$
```

```
          A = SQRT(1.0 - INC)$     B = 1.0 - INC $
                W     =   ESTFP = 0.0 $
          FOR I = (1,1,M) DO BEGIN
B8             IT = IT(I)$     ID = DD(I)$
               B1 = ID NEQ 0 $   B2 = IT NEQ 0 $
               JD(I) = ID $    JT(I) = IT$
               FOR J = (1,1,I-1) DO BEGIN
B9                   IF B2 AND IT EQL JT(J) THEN B2 = FALSE $
E9                   IF B1 AND ID EQL JD(J) THEN B1 = FALSE $     END$
               IF B1 AND B2 THEN BEGIN
B10                  ALPHA(ID) = A*ALPHA(ID)$
E10                  BETA(IT)  = A*BETA(IT)$ END ELSE
               IF B1 THEN BEGIN
B11                  DS = B*ALPHA(ID)$    TS = RAL*BETA(IT)*T(I)$
                     IF DS LEQ TS THEN ALPHA(  ID) = (TS + ALPHA(ID)*D(I))/2.0
E11                          ELSE ALPHA(ID) = DS $    END  ELSE
               IF B2 THEN BEGIN
B12                  TS = B*BETA(ID)$    DS = ALPHA(ID)*D(I)/RAU$
                     IF TS GEQ DS THEN BETA(IT) = (DS + BETA(IT)*T(I))/2.0
E12                          ELSE BETA(IT) = TS $     END$
               ALPH = ALPHA(ID)*D(I)$
               BET =   BETA(IT)*T(I)$
               W = W + ALPH*BET*WF(I)$
               ESTFP = ESTFP + 1.0/(RAU-ALPH/BET) + 1.0/(ALPH/BET - RAL)
                    + 1.0/(DMAX - ALPH) + 1.0/( TMAX - BET) + 1.0/ALPH $
E8                                        END$
B13       FOR K = (1,1,KT) DO FOR J = (1,1,N) DO BEGIN        TS = U(J,K)/B$
                     IF TS LSS UUP AND TS GTR ULO THEN U(J,K) = TS$
               ESTFP = ESTFP + KUZ/(UUP - U(J,K) ) + KUZ/(U(J,K) - ULO)$
E13            END$
          LAB..
          JJ = 0 $
          FOR I = (1,1,M) DO BEGIN
B14            IT = IT(I)$    ID = DD(I) $
               ALPH = ALPHA(ID)*D(I)$    BET = BETA(IT)*T(I)$
               BS = BUC(I)*(ALPH*ALPH + BET*BET)$
               CS = 0.40*E(I)*BET/ALPH $
          LAC..
               FOR K =(1,1,KT) DO BEGIN
B15                  AA = 0.0 $              FOR J = 1,2,3 DO
                             AA = AA + EGL2(I,J)* (U(SR(I,J),K)-U( SR(I,J+3),K))$
               DS = MAX(SMIN,-BS,-CS)$
               AA = 1.0001*AA $
               IF AA LEQ DS OR AA GEQ SMAX(I) THEN BEGIN
B16                  TS = 0.96*MAX(DS/AA,SMAX(I)/AA)$
                     FOR J = (1,1,6) DO U(SR(I,J),K) = TS*U( SR(I,J),K)$
                     JJ = 1 $
E16                  GO TO LAC $     END$
                     ESTFP = ESTFP + 50000*( 1.0/(SMAX(I) - AA) +
                             1.0/(AA-SMIN(I)) + 1.0/( CZ1*( AA + BS)) +
                             1.0/(AA + CS ))$
E15                          END$
E14                  END$
B17  E17  IF JJ NEQ 0 THEN BEGIN   WRITE (J)$ GO TO LAB $ END$
```

```
                      WRITE(ALPHA,BETA,U,ESTEP)$
                      WO = 1.005*WS
                      WRITE (WO)$
E7                    ENDS
END BLOCK 6   LEVEL 4
                      COMMENT IF THIS IS THE FIRST TIME THROUGH THE PROCEDURE READ IN THE DATA
                           AND SET UP THE CONSTANTS WHICH WILL BE USED IN THE SYNTHESIS $
                      IF B1 THEN BEGIN
B18                   READ (RHO)$
                           READ (KUZ)$
                      READ (B5,B7)$
                      READ (B9)$
                           PI = 3.1415927 $
                           LL(1,1) = 1.0 $           LL(1,2) = -1.0 $
                           LL(2,1) = -1.0 $          LL(2,2) = 1.0 $
                           FOR I = (1,1,M) DO READ (LISTA)       $
                           K = 0 $
                           FOR I = (1,1,NN) DO BEGIN
B19                             READ (LISTH)$
                                J = 3*J $
B20     E20                     IF V NEQ 0 THEN BEGIN K = K + 1 $ UU(J-2) = K $ ENDS
B21     E21                     IF B NEQ 0 THEN BEGIN K = K + 1 $ UU(J-1) = K$ ENDS
B22     E22                     IF L NEQ 0 THEN BEGIN K = K + 1 $ UU(J    ) = K $ ENDS
E19                                                                 ENDS
                           FOR K = (1,1,KT) DO BEGIN
B23                             READ (J) $
                                FOR I = (1,1,J) DO BEGIN
B24                                  READ (L,P1, P2, P3) $
                                     V = 3*L$
                                     P( UU( V - 2), K) = P1 $
                                     P( UU( V - 1), K) = P2 $
E24                                  P( UU(V)      , K) = P3$                   ENDS
E23                                                                ENDS
                           FOR I = (0,1,NDV) DO READ( ALPHA(I) )$
                           FOR I = (0,1,NTV) DO READ( BETA(I))$
                      FOR I = (1,1,M) DO BEGIN
B25                        PT1 = PT(I,1) $
                           PT2 = PT(I,2)$
                           G1 = COORD(PT1,1) - COORD(PT2,1)$
                           G2 = COORD(PT1,2) - COORD(PT2,2)$
                           G3 = COORD(PT1,3) - COORD(PT2,3) $
                           LEN(I) = SQRT( G1*G1 + G2*G2 + G3*G3)$
                           KCONST(I) = PI*E(I)/LEN(I)$
                           G1 = G1/LEN(I)$
                           G2 = G2/LEN(I)$
                           G3 = G3/LEN(I)$
                           DRC(I,1) = G1$
                           DRC(I,2) = G2 $
                           DRC(I,3) = G3 $
                           CON1 = G1*G1*KCONST(I)$
                           CON2 = G2*G2*KCONST(I)$
                           CON3 = G3*G3*KCONST(I)$
                           CON4 = G1*G2*KCONST(I)$
                           CON5 = G2*G3*KCONST(I)$
```

```
          CONB = G3*G1*KCONST(I)$
          KLOCAL(I,1,1) = CON1 $
          KLOCAL(I,2,2) = CON2 $
          KLOCAL(I,3,3) = CON3 $
          KLOCAL(I,1,2) =KLOCAL I,2,1) = CON4 $
          KLOCAL(I,1,3) = KLOCAL(I,3,1) = CON6 $
          KLOCAL(I,2,3) = KLOCAL(I,3,2) = CON5 $
E25       END$
          BB = FALSE $
          READ (SFE,SFC)$
          READ (RAU,RAL)$
          G1 = PI*PI $
          W = 0 $
          FOR I = (1,1,M) DO BEGIN
B26           KCONST(I) = ALPHA( DD(I) )*BETA( TT(I) )*D(I)*T(I)$
            PT1 = PT(I,1) $
            PT2 = PT(I,2) $
            FOR L = 1,2,3 DO BEGIN
B27               SR(I,L) = UU(3*PT1 - 3 + L)$
                  SR(I,L+3)=UU(3*PT2 - 3 + L ) $
E27                              END$
            EBL(I) = E(I)/LEN(I) $
            FOR J = 1,2,3 DO EBL2(I,J) = EBL(I)*DRC(I,J)   $
            WF(I) = PI*RHO*LEN(I)$
            W = W + WF(I)*KCONST(I) $
            BUC(I) = G1*EBL(I)/(8*SFE*LF (I) ) $
E26                      END$
          W0 = 1.0005*W $
          WA = WB = W $
          BB = FALSE $
          NKT = N*KI $
          NKTD = NKI + NDV $
          NKTDT = NKTD + NTV $
          FOR I = (1,1,NKT) DO X(I) = 0.0 $
          FOR I = (1,1,NTV) DO X(NKTD + I) = BETA(I) $
          FOR I = (1,1,NDV ) DO X(NKT + I ) = ALPHA(I) $
          C21 = 125 $
          READ ( UUP , ULU)$
          READ (DMAX,TMAX)$
          READ (MXE )$
          READ        (    NU , WM , MJ , RNU , NUDIV )$
          FOR I = (1,1,NKTD)) DO TSCALE(I) = 1.0 $
          THG = NU $
          FOR I = (1,1,NKTDT) DO X(I) = X(I)/TSCALE(I) $
          READ (I)$
          IF I GTR 0 THEN READ (WA)$
          IF I GTR  0 THEN READ( NU,THG,WM,W0,X,TSCALE          )$
                IF B9 THEN BEGIN
B28       WRITE (FR11)$
          FOR I = (1,1,M) DO WRITE (FR12,L55)$
          WRITE (FR13,L56)$
          WRITE (FR14)$
          FOR I = (1,1,NN) DO WRITE (FR15,L57)$
          WRITE (FR16)$
```

```
         FOR K = (1,1,KT) DO BEGIN
B29           WRITE(FR17,K)$
E29           FOR I = (1,1,NN) DO WRITE (FR18,LSA)$      END$
E28                                END$
         CZ1I = 1.0/CZ1$
         READ (INC)$
         WRITE (EJ)$
E18      END$
         LI4..
         B10 = FALSE $
         NU1 = NU $
         NU2 = 50000*NU $
         IF CHEK EQL 4 OR CHEK EQL 43 THEN WRITE (FOR I = (1,1,NKID1) DO
                    X(I)*TSCALE(I)          )$
B30      FOR K = (1,1,KI) DO BEGIN KN = (K-1) * N $
              FOR I = (1,1,N) DO
                    U(I,K) = X(KN + I )*TSCALE( KN + I) $
E30                                                   END$
         FOR I = (NKT + 1,1,NKTD) DO ALPHA(I-NKT) = X(I)*TSCALE(I)$
         FOR I = (NKTD +1,1,NKTDT) DO BETA(I-NKTD)= X(I)*TSCALE(I)$
         B4 = B6 = FALSE $
         LI7..
         IF B3 THEN FOR I = (1,1,NKID1) DO DELX(I) = 0.0 $
         IF CHEK EQL 43 THEN BEGIN
B31           B4 = TRUE $
              FOR I = (1,1,NKID1 ) DO BEGIN
B32                 TSKALE(I) = TSCALE(I) $
                    TSCALE(I) = 0.0 $
E32                                 END$
E31                       END$

         ALT = 0 $
         W = 0.0 $
         R = 0.0 $
         PHEE = 0.0 $
         COMMENT   COMPUTE THE WEIGHT OF THE DESIGN $
         FOR I = (1,1,M) DO BEGIN
B33           IT = TT(I) $
              ID = DD(I) $
              ALPH = ALPHA(ID)*A(I)$
              BET = BETA(IT)*T(I) $
              KCONST(I) = ALPH*BET $
              W = W + KCONST(I)*WF(I) $
E33           END$
         COMMENT  CALCULATE THE RESIDUALS FOR THE DESIGN $
         AA = BB = 0.0 $
         FOR K = (1,1,KT) DO BEGIN
B34           U(0,K) = 0.0 $
              FOR L = (1,1,N) DO RV(L) = 0.0 $
              FOR I = (1,1,M) DO BEGIN
B35                 FOR L = (1,1,6) DO LOCU(L) = U(SR(I,L),K)$
                    FOR V = (1,1,3) DO BEGIN
B36                      TS = KCONST(I)*(KLOCAL(I,V,1)*(LOCU(1) - LOCU(4)) +
                         KLOCAL(I,V,2)*(LOCU(2) - LOCU(5)) + KLOCAL(I,V,3)*(
```

```
                         LOCO(3) - LOCO(6))     )$
                         J = SR(I,V) $
                         L = SR( I,V+3)$
                         RV(J) = RV(J) + TS $
                         RV(L) = RV(L) - TS $
E36                                             ENDS
E35                                         ENDS
                   FOR L = (1,1,N) DO BEGIN
B37                      TS = RVS(L,K) = RV(L) - P(L,K) $
                         R = R + TS*TS $
                         BB = MIN(TS,BB)$
                         AA = MAX( TS,AA)$
E37                                        ENDS
E34                          ENDS
      COMMENT  END THE COMPUTATIONS OF THE RESIDUALS $
      R = R*RNO/1000000 $

           MJ = MAX(  (0.025 - 0.020*W/WA)*MXE,5.0) $
           IF NOT B5 THEN MJ = MJ/1.2 $
      COMMENT  SEE IF THE RESIDUALS ARE SATISFIED $
           RESIDUALS = MAX(AA,-BB) LEQ MJ $
           C22 = N*KI*MJ*MJ*1.0R-6 $
      LI9..
           IF RESIDUALS T  N BEGIN
B38        IF NOT B6 AND     NEW + THEN BEGIN
B39              WRITE (' ...UALS MAY BE SATISFIED')$
                 B10 = TRUE $
FR    E39        GO TO LI5 $     ENDS
                 EXTRAP(FALSE,MO,R,BB)$
                 IF B5 AND (  WO - W)/W LSS 0.5*INC THEN WM = 2.0*WM $
                 WA = W $
                         WRITE ( EJ)$
                         WRITE (FORA)$
                         WRITE (FR1,LS1)$
                         WRITE ('      DESIGN WEIGHT IS')$
                         WRITE (F1A,W)$
                         WRITE (  ' ',' ',' ',' ','RESIDUALS ARE SATISFIED')$
                         WRITE (FORE,MJ)$
                         IF NOT B6 THEN  BEGIN
B40                      WRITE (FORB,LS2)$
B41                      FOR I =(1,1,NN) DO BEGIN  WRITE (  FR2,1)$
                                                  J = 1 $ WRITE (FR2X,LS3)$
                                                  J = 2 $ WRITE (FR2Y,LS3)$
E41                                               J = 3 $ WRITE (FR2Z,LS3)$ ENDS
E40                                          ENDS
                         WRITE ( FORC,LS2)$
B42                      FOR I =(1,1,NN) DO BEGIN  WRITE (  FR2,1)$
                                                  J = 1 $ WRITE (FR4X,LS4)$
                                                  J = 2 $ WRITE (FR4Y,LS4)$
E42                                               J = 3 $ WRITE (FR4Z,LS4)$ ENDS
                   FOR I = (1,1,M) DO BEGIN
B43                      CON1 = ALPHA( DD(I) )*D(I)$
                         CON2 =  BETA( TI(I) )*T(I)$
                         STRESS(I, 0) = BUC(I)*( CON1*CON1 + CON2*CON2)$
```

```
                       STRESS(I,-1) = 0.40*E(I)*CON2/CON1$
                       FOR K = (1,1,KT) DO BEGIN
B44                        AA = 0.0 $
                           FOR J = 1,2,3 DO
                       AA = AA + EGL2(I,J)*(U(SR(I,J),K)-U(SR(I,J+3),K) )$
E44                        STRESS(I,K) = AA$    END$
E43                    END$
              MODIF(U,ALPHA,BETA,W0,INC,M,G1)$
                  NU = MIN(NU, NUDIV*NUDIV*C22/( G1 + 60.0)     )$
                  NU = SQRT( NU*THG )$
                  THG = NU $
                       WRITE (FORD,NU,WM,W0)$
                       WRITE         (EJ)$
                  FOR I = (1,1,NKTOT) DO TSCALE(I) = 1.0 $
B45               FOR K = (1,1,KT) DO BEGIN    KN =(K-1)*N$
E45                  FOR J = (1,1,N) DO X(J + KN) = U(J,K) $ END$
                  FOR I = (1+NKT,1,NKTU) DO X(I) = ALPHA(I- NKT)$
                  FOR I = (1+NKTU ,1,NKTOT) DO X(I) = BETA(I- NKTD) $
                  B3 = B4 = B5 = B6 = TRUE $
                 CHEK = 43 $
                 GO TO LI7 $
E38           END ELSE IF CHEK EQL 4 THEN BEGIN
B46                    WRITE (EJ)$
                       WRITE (FORA)$
                       WRITE (FR1,LS1)$
                       WRITE ('PRESENT WEIGHT IS')$
                       WRITE (F1A,W)$
                       WRITE ( ' ',' ',' ',' ','RESIDUALS ARE NOT SATISFIED')$
                       WRITE (FORE,MJ)$
                       WRITE (FORB,LS2)$
B47                    FOR I =(1,1,NN) DO BEGIN  WRITE (  FR2,I)$
                                                  J = 1 $ WRITE (FR2X,LS3)$
                                                  J = 2 $ WRITE (FR2Y,LS3)$
E47                                               J = 3 $ WRITE (FR2Z,LS3)$ END$
                  EXTRAP(TRUE,NU,R,B5)$
FR                IF B6 THEN GO TO RETREAT $
                  NU = NU/NUDIV $
                       WRITE (FR3,NU)$
                       WRITE (EJ)$
E46                                          END$
                  NU1 = NU $
                  NU2 = 50000*NU $
          LI5..
          MESS = W0*NU*WM $
          RNUL = RNU/500000$
          COMMENT CHECK THE WEIGHT CONSTRAINT $
          W = W0 - W $
          IF W LEQ 0 THEN BEGIN
B48           ALT = ' $
FR    E48     GO TO RETREAT $      END$
          IF     THEN
          C     NT BEGIN THE COMPUTATIONS FOR THE FIRST AND SECOND PARTIAL DERIV.
              OF THE RESIDUAL $
          FOR K = (1,1,KT) DO BEGIN
```

```
B49         KN = N*(K-1) $
            FOR I = (1,1,M) DO BEGIN
B50              IT = TT(1) $
                 ID = DD(1) $
                 BET = BETA(IT)*T(I)$
                 ALPH = ALPHA(ID)*D(I)$
                 FOR L = (1,1,6) DO BEGIN
B51                   J = SR(I,L) $
                      PUR(L) = U(J,K)$
E51                   LOCU(L) = RVS(J,K)$      ENDS
                 G1 = G2 = 0.0 $
                 FOR L = (1,1,3) DO BEGIN
B52                   TS = TS1 = BB = 0.0 $
                      FOR J = (1,1,3) DO BEGIN
B53                        IF KLOCAL(I,L,J) NEQ 0.0 THEN BEGIN
B54                        CON7 = KLOCAL(I,L,J)*PUR(J)$
                           CON8 = KLOCAL(I,L,J)*PUR(J + 3)$
                           AA = CON7 - CON8$
                           TS = TS + AA $
                           G1 = G1 + CON7*CON7 + CON8*CON8$
                           BB = BB + KLOCAL(I,L,J) *(LOCU(J) - LOCU(J+3))$
E54                        ENDS
E53                                                ENDS
                      TS1 = ALPH*TSS
                      TS  =  BET*TS$
                      BB = KCONST(I)*BB $
                      AA = LOCU(L) $
                      G3 = LOCU(L+3) $
                      PLOC(L) = TS*AA $
                      PLOC(L+3) = -TS*G3 $
                      PLC(L) = TS1*AA $
                      PLC(L+3) = -TS1*G3 $
                      KLC(L) = BB $
                      KLC(L+3) = -BB $
E52                                                ENDS
                      G2 = ALPH*ALPH*G1 $
                      G1 =  BET* BET*G1 $
                 FOR L = (1,1,6) DO BEGIN
B55                   IF IT NEQ 0 THEN
                      DELX( NKTD + IT) = DELX( NKTD + IT) + PLC(L)*RNU5 $
                      IF ID NEQ 0 THEN
                      DELX(  NKT + ID) = DELX(  NKT + ID) + PLOC(L)*RNU5 $
                      J = SR(I,L)$
                      IF J NEQ 0 THEN
                      DELX(   KN + J ) = DELX(   KN +  J)  + KLC(L)*RNU5 $
E55                                                 ENDS
                                                           IF B4 THEN BEGIN
B56              IF IT NEQ 0 THEN
                 TSCALE( NKTD + IT) = TSCALE( NKTD + IT) + RNU*G2/500000 $
                 IF ID NEQ 0 THEN
                 TSCALE( NKT + ID) = TSCALE( NKT + ID) + RNU*G1/500000$
E56                                                                ENDS
E50                                                 ENDS
E49                              ENDS
```

```
          IF B4 THEN BEGIN
B57       FOR I = (0,1,N) DO LITTLE(I) = 0.0 $
          FOR I = (1,1,M) DO BEGIN
B58            FOR L = (1,1,3) DO BEGIN
B59                 AA = KLOCAL(I,L,L) * KCONST(I),
                    U1 = SR(I,L) $
                    U2 = SR(I,L+3)$
                    LITTLE(U1) = LITTLE(U1) + AA $
                    LITTLE(U2) = LITTLE(U2) + AA $
E59                                                                  END$
E58                       END$
                    FOR I = (1,1,N) DO LITTLE(I) = LITTLE(I)*LITTLE(I)*RNU5 $
          FOR K = (1,1,KT) DO BEGIN
B60            KN = N*(K-1) $
               FOR I = (1,1,N) DO
                    TSCALE( KN + I) = TSCALE( KN + I) + LITTLE(I)$
E60                                       END$
E57                            END$
          ALT = 0 $
          IF NOT B5 THEN BEGIN
B61            MESS = 0.0 $
               FOR I = (NKT+1,1,NKIDT) DO      DELX(I) = 0.0 $
FR             GO TO LIB $
E61                 END$
          COMMENT CHECK THE DISPLACEMENT CONSTRAINTS $
          FOR K = (1,1,KT) DO BEGIN
B62            V = (K - 1)*N$
               FOR J = (1,1,N) DO BEGIN
B63            G1 = UUP - U(J,K)$
               G2 = U(J,K) - ULO$
               G1S = G1*G1 $
               G2S = G2*G2 $
               G1C = G1S*G1 $
               G2C = G2S*G2 $
               IF G1C LEQ 0.0 OR G2C LEQ 0.0 THEN BEGIN
B64                 ALT = 3 $
FR   E64            GO TO RETREAT $      END$
               PHEE = PHEE + KUZ*NU*(1.0/G1 + 1.0/G2)$
               DELX(J+V) = DELX(J+V) + KUZ*NU*(1.0/G1S - 1.0/G2S)$
               IF B4 THEN
                    TSCALE(J+V) = TSCALE(J+V) + 2*KUZ*NU*(1.0/G1C + 1.0/G2C)$
E63                                                 END$
E62                                 END$
          FOR K = (1,1,KT) DO U(J,K) = 0.0 $
          FOR I = (1,1,M) DO BEGIN
B65            IT = TT(I) $
               ID = DD(I) $
               TS = TS1 = 0.0 $
               AAA = BBB = 0.0 $
               NU = NU2 $
               CON1 = ALPHA(ID)*D(I)$
               CON2 = BETA(IT)*T(I)$
               CON5 = RUC(I)* ( CON1*CON1 + CON2*CON2) $
               CON4 = 0.40*E(I)*CON2/CON1 $
```

```
              C1 = 2*BUC(I)*CON1$
              C2 = 2*BUC(I)*CON2 $
              C3 = CON4/CON1$
              C4 = CON4/CON2 $
              FOR J = (1,1,3) DO BEGIN
B66                G1 = EQL2(I,J) $
                   PARTU(J) = G1 $
E66                PARTU(J+3) = -G1 $          END$
              FOR K = (1,1,KT) DO BEGIN
B67                AA = 0.0 $
                   FOR J = 1,2,3 DO
                        AA = AA + EQL2(I,J)*( UC SR(I,J),K) - UC SR(I,J +3),K)) $
                   SIG = AA $
                   G1 = SMAX(I) - SIG $
                   G2 = SIG - SMIN(I) $
                   G3 = SIG + CON3 $
                   G4 = SIG + CON4 $
                   G1 = SOS(G1,-13)$
                   G2 = SOS(G2,-13)$
                   G3 = SOS(G3,-13)$
                   G4 = SOS(G4,-13)$
                   IF G1 LEQ 0.0 OR G2 LEQ 0.0 OR G3 LEQ 0.0 OR G4 LEQ 0.0
                        THEN BEGIN
B68                     ALT = 2 $
                        NO = NO1$
FR     END              GO TO RETRACTS END$
                   G11 = 1.0/G1 $             G21 = 1.0/G2$
                   G31 = 1.0/G3$              G41 = 1.0/G4$
                   G15 = G11*G1.              G1C = G1I*G15$
                   G25 = G21*G21$             G2C = G2I*G25$
                   G35 = G31*G31*C7I1$        G3C = G3I*G35$
                   G45 = G41*G41$             G4C = G4I*G45$
                   G31 = G31*C2I1$
                   PREF = PREF + NO*(G1I + G2I + G3I + G4I)$
                   IF B3 THEN BEGIN
              COMMENT : GRADIENT FOR STRESS RESTRAINTS $
B69                     CONB = NO*(G15 - G25 - G35 - G45)$
                        IF B4 THEN CON4 = NO*( G1C + G2C + G3C + G4C )$
                        H = N*(K - 1) $
                        FOR J = (1,1,3) DO BEGIN
B70                          J1 = SR(I,J) + H $
                             IF J1 NEQ 0 THEN BEGIN
B71                               DELX(J1) = DELX(J1) + PARTU(J)*CONB $
                                  IF B4 THEN
                                  TSCALE(J1) = TSCALE(J1) + 2*PARTU(J)*
                                       PARTU(J)*CON4 $
E71                                         END$
E70                                                       END$
                        TS = TS + NO*(C3*G45 - C1*G35)$
                        TS1 = TS1 + NO*( -C2*G3. - C4*G45)$
                        IF B4 THEN BEGIN
B72                          AAA = AAA + NO*2.0*( C1*C1*G3C - BUC(I)*G35 +
                                  C3*C3 * G4C - C3*G45/CON1    )$
                             BBB = BBB + NO*2.0*( C2*C2*G3C - BUC(I)*G35   +
```

```
E72                  C4*C4*G4C)$        END$
E69  C               END OF GRADIENT OF STRESS RESTRAINTS $
E67              END$
          NU = NU1 $
          COMMENT CHECK THE SIDE CONSTRAINTS$
          G3 = DMAX - CON1 $       G3S = G3*G3$               G3C = G3S*G3$
          G4 = TMAX - CON2$        G4S = G4*G4$               G4C = G4S*G4 $
          IF G3C LEQ 0.0 OR G4C LEQ 0.0 THEN BEGIN
B73           ALT = 4 $
FR   E73      GO TO RETREAT $ END$
          PHEE = PHEE + NU*(1.0/G3 + 1.0/G4)$
          TS = TS + NU/G3S $
          TS1=TS1+NU/G4S$
          IF B4 THEN       BEGIN
B74           AAA = AAA + 2.0*NU/G3C$
              BBB = BBB + 2.0*NU/G4C $
E74                 END$
          CON3 = CON1/CON2 $
          G1 = CON3 - RAL $
          G2 = RAU - CON3 $
          G1 = SOS(G1,-13)$       G2 = SOS(G2,-13)$
          IF  G1 LEQ 0.0 OR G2 LEQ 0.0 THEN BEGIN
B75           ALT = 1 $
FR   E75      GO TO RETREAT $                    END$
          CON3 = CON3/CON2 $
          CON4 = CON3 * CON3             $
          CON5 = CON3/CON2 $
          G1S = G1*G1 $
          G2S = G2*G2 $
          G1C = G1*G1S $
          G2C = G2*G2S $
          PHEE = PHEE + NU*( 1.0/G1 + 1.0/G2) $
          TS = TS + NU*( 1/G2S - 1/G1S)/CON2 $
          TS1=TS1+ NU*(1/G1S - 1/G2S)*CON3 $
          IF B4 THEN BEGIN
B76           AAA = AAA + NU*(2/G1C + 2/G2C)/(CON2*CON2)$
              BBB = BBB + NU*( (2/G2S - 2/G1S)*CON5 + (2/G1C + 2/G2C)*CON4)$
E76                       END$
          CON1 = SOS(CON1,-13)$
B77  FR    E77  IF CON1 LEQ 0.0 THEN BEGIN ALT = 3 $  GO TO RETREAT$ END$
          PHEE = PHEE + NU/CON1 $
          IF B3 THEN TS = TS - NU/( CON1*CON1)$
          IF B4 THEN AAA = AAA + 2*NU/( CON1*CON1*CON1)$
          CON6 = W*W $
          CON7 = CON6*W $
          G1 = MESS * WF(I)/CON6 $
          ALPH = CON1 $
          BET = CON2 $
          TS = TS + G1*BET $
          TS1= TS1 + G1*ALPH $
          IF B4 THEN BEGIN
B78           CON9 = 2*MESS*WF(I)*WF(I)/CON7 $
              IF ID NEQ 0 THEN
                  TSCALE( NKT+ID ) = TSCALE( NKT + ID) + AAA +CON9*BET*BET$
```

```
                IF IT NEQ 0 THEN
                      TSCALE( NKTD+IT) = TSCALE( NKTD+IT) +BBB + CON9*ALPH*ALPH $
E78           END$
           IF ID NEQ 0 THEN  DELX(NKT + ID) = DELX(NKT + ID) + TS $
           IF IT NEQ 0 THEN DELX(NKTD + IT ) = DELX(NKTD + IT) + TS1 $
E65                END$
           SAVPHI = PHEE $
           LI8..
           IF B10 THEN BEGIN
B79              CHEK = 4 $
                 W = W0 - W $
                 B10 = FALSE $
E79              GO TO LI9$   END$
           IF B4 THEN BEGIN
B80              FOR K = (1,1,NKTDT) DO X(K) = X(K)*TSKALE(K) $
                 IF NOT B5 THEN FOR I = (NKT+1,1,NKTDT) DO TSCALE(I) = 1.0$
                 FOR I  = (1,1,NKTDT) DO TSCALE(I) = 1.0/SQRT( ABS(TSCALE(I) ) ) $
                 G1 = MAX( TSCALE)$
                 FOR I = (1,1,NKTDT) DO TSCALE(I) = TSCALE(I)/G1 $
                 IF NOT B5 THEN FOR I = (1 + NKT,1,NKTDT) DO TSCALE(I) = 1.0 $
FR               IF CHEK EQL 4  OR B6 THEN GO TO LSR$
           FOR I = (1,1,NKTDT) DO
                 IF ABS( (TSCALE(I) - TSKALE(I))/TSKALE(I) ) GTR 0.60 THEN GO TO
FR                                   LSR$
                 FOR I =(1,1,NKTDT) DO TSCALE(I) = TSKALE(I)$
                 DELX(0) = 8.0 $
                 LSR..
           WRITE ('**--**--*--*--**--**--*--*---*---*----*---',TSCALE)$
                 FOR I = (1,1,NKTDT) DO X(I) = X(I)/TSCALE(I) $
E80              END$
           IF CHEK EQL 43 OR CHEK EQL 4 THEN WRITE (F1A, NU,THG,WM,W0,X,TSCALE)$
           CTJ = CTJ + 1 $
           IF CTJ GEQ 35 THEN BEGIN
B81              CTJ = 0 $
                 WRITE ('**')$
                 WRITE (F1A,R,W0-W)$
E81              END$
           RETREAT..
           IF B6 THEN BEGIN
B82              WRITE (EJ)$
                 WRITE ('STRESSES IN TRUSS MEMBERS')$
                 V = M//5 $
                 FOR I = (1,1,1+V) DO BEGIN
B83                    IF I LEQ V THEN L = 5 ELSE L = M -5*V$
                       B = 5*(I-1)$
                       WRITE (FR5,FOR J = (1,1,L) DO B + J)$
                       WRITE (FR6, FOR J = (1,1,L) DO SMAX(B+J) )$
                       WRITE (FR7, FOR J = (1,1,L) DO SMIN(B+J) )$
                       WRITE (FR8, FOR J = (1,1,L) DO -STRESS(B+J,0))$
                       WRITE (FR9, FOR J = (1,1,L) DO - STRESS(B+J,-1))$
                       WRITE ('     LOAD CONDITION')$
                       FOR K = (1,1,K1) DO
                             WRITE (FR10,K,FOR J = (1,1,L) DO STRESS( B+J,K))$
E83                                                        END$
```

```
                       WRITE (EJ)$
E82                                                ENDS
                  IF B6 THEN ALT = 8 $
                  IF B8 THEN BEGIN
B84                    WRITE ('THE LAST BALANCED DESIGN WILL BE ACCEPTED AS THE OPTIMUM')$
                       DELX(0) = 101.0 $
E84                 ENDS
                  IF B6 AND B7 THEN BEGIN
B85                    WRITE (PUNCH,NU,THG,WM,WO,X,TSCALE)$
E85                    WRITE (      NU,THG,WM,WO,X,TSCALE)$           ENDS
                       IF ALT EQL 8 OR ALT EQL 0 THEN
                  FOR I = (1,1,NKTOT) DO BEGIN
B86                    DELX(I) = DELX(I)*TSCALE(I)$
E86                    ENDS
                  ROW = ALT $
                  PHEE = PHEE + K + MESS / W $
E2                ENDS
END BLOCK 3   LEVEL 3
E1                ENDS
END BLOCK 2   LEVEL 2
END BLOCK 1   LEVEL 1

COMPILATION COMPLETED IN   34.76 SECONDS
```

```
  ALG     FLKE
COMPILED BY CASE 1107 ALGOL 60 (FAST VERSION)  DATED  DECEMBER 14, 1967
THIS COMPILATION WAS DONE ON 09 APR 68 AT 14:51:42
B0                              EXTERNAL PROCEDURE FP $
BLOCK 1      LEVEL 1
                                INTEGER ND,KT,N,M,NN,NDV,NTV$
                                READ (KT,N,M,NN,NDV,NTV)$
                                ND = N $
                                N = ND*KT + NDV + NTV $
                                BEGIN
                                COMMENT FLETCHER - REEVES $
B1                              PROCEDURE SGEN1(N,S,GX,GXEM)$
BLOCK 2      LEVEL 2
BLOCK 3      LEVEL 3
                                      REAL ARRAY S,GX,GXEM $
                                      INTEGER N $
                                      BEGIN
B2                                    REAL BETA,BE,TA$
                                      INTEGER I$
                                      BE=TA = 0.0 $
                                      FOR I = (1,1,N) DO BEGIN
B3                                          BE = BE + GXEM(I)*GXEM(I)$
                                            TA = TA + GX(I)*GX(I)$
E3                                                          ENDS
                                      BETA = BE/TA $
                                      FOR I = (1,1,N) DO S(I) = GXEM(I) + BETA*S(I)$
E2                                    END$
END BLOCK 3   LEVEL 3
                                REAL ARRAY A,U,GXEM,XEM,GA,GB,S,TGA,XX(0:N)$
                                REAL D1,D2,D3,H1,H2,H3,  FX, EPA,EPB,EG, F1, T, FEM, GG,
                                      TS, A, B, FA, FB, DAS, GBS, GS, AGXEM, AG, TEST,
                                      H, GXS, FEST, TIME, CTIME, QG , TO, TM, TZ, TC,
                                      T2, G2, TMIN, TSA, TEA, TA, Z, W ,TGAS            $
                                LOCAL LABEL L5A,UPTOP $
                                INTEGER Q,I, RELOCS, ROW, CHEK, CTRA, CTRB, CTRC , CTRD, CTRE $
                                BOOLEAN B1,B2,B3,B4,B5,B6,B7      $
                                FORMAT FORA(     8R16.8,A1)$
                                TIME = CTIME = CLOK $
                                CHEK = 43$
                                B1 = B2 = B3 = TRUE $
                                B4 = FALSE $
                                D1 = D2 = D3 = H1 =  H2 = H3 = 0.0 $
                                EPB = 1.0&-4 $
                                EG = 0.0 $
                                QG = 1.0 $
                                FP( ROW,X,G,F1,B1,B2,B3,CHEK )$
                                READ (CTRC)$
                                READ ( B5 )$
                                CTRB = CTRC $
                                Q = 0$
                                B1 = FALSE $
                                WRITE (       CHEK,' ','A', X, ' ', F1, ' ', G)$
```

```
        UPTOP:
        TS = 0.0 $
        FOR I = (1,1,N) DO BEGIN
B4            S(I) = G(I) $
              TS = TS + S(I)*S(I)$
E4                                  ENDS
        S(0) = SQRT( TS)$


        TOP:
        FEST = 0.0 $
        CTRA = 0 $
        CHEK = 1 $
        Q = 0 $


        L2:
        IF NOT B5 THEN
              IF CTRA GEQ CTRB THEN BEGIN
B5                  CHEK = 43 $
                    FP ( ROW,X,G,F1,B1,B2,B3,CHEK)$
                    IF G(0) EQL 8.0 THEN BEGIN
B6                        G(0) = 0.0 $
                          CTRB = 2*CTRB $
E6                        GO TO TOP $    ENDS
                    CTRD = CTRD + 1 $
                    IF  MOD( CTRD,3) EQL 0 THEN BEGIN
B7                        CTRD = 0 $
                          CTRB = CTRB + 1 $
E7                        GO TO TOP $ ENDS
                    GO TO UPTOP $
E5                                                ENDS


        TS = 0.0 $
        FOR I = (1,1,N) DO
              TS = TS + G(I)*S(I) $
        GG = TS $
        IF GG LSS 0.0 THEN BEGIN
B8            FOR I = (1,1,N) DO S(I) = G(I)$
              WRITE ('S IS REPLACED BY G BECAUSE G DOT S IS NEGATIVE')$
              GO TO UPTOP $
E8                      ENDS


        L4:
        RELOCS = 0 $
        TS = F1 - FEST $
```

```
          IF TS LSS 0.0 THEN BEGIN
B9             FEST = FEST + 2.0*TS $
               GO TO L4$
E9                         END$
          GAS = - GG $
          IF QC EQL 0.0 THEN QQ = 1.0 $
          T = 2.0*QQ*TS/GG $
          TG = T $
          B4 = FALSE $
          CTRA = CTRA + 1 $
          TMIN = 0.0 $
          T2 = 0.0 $
          G2 = GAS $
          WRITE (G2)$
          B7 = FALSE $



          UPPER:
          H = T $
          B6 = TRUE$



          INCREASE:
          IF B7 AND H GEQ TMIN THEN BEGIN
B10            ROW = 1 $
FR   E10       GO TO MID$     END$
          FOR I = (1,1,N) DO XX(I) = X(I) - H*S(I)$
          B = H $
          FP(ROW,XX,GB,FB,B1,B2,B3,CHEK)$



          MID:
          WRITE (H,ROW,FB)$
          IF ROW NEQ 0 THEN BEGIN
B11            IF ROW EQL 8 THEN BEGIN
B12                 WRITE ('RESIDUALS WERE SATISFIED IN PART 1 OF LIN SEARCH')$
                    F1 = FB $
                    B5 = FALSE $
                    CTRD = 0$
                    CTRB = CTRC $
                    FOR I = (1,1,N) DO BEGIN
B13                      G(I) = GB(I)$
E13                      X(I) = XA(I)$   END$
                  GO TO UPTOP $
E12                              END$
               B7 = TRUE $
               IF TMIN EQL 0.0 THEN TMIN = H $
               IF H LSS TMIN THEN TMIN = H $
               IF B6 THEN BEGIN
B14                 T = T/8.0 $
E14                 GO TO UPPER$ END ELSE BEGIN
```

```
B15                      H = TA + (H - TA)/2.0 $
E15                      GO TO INCREASE$          END$
E11                                  END$
               TSA = TS = 0.0 $
               FOR I = (1,1,N) DO BEGIN
B16                 TS = TS - GB(I)*S(I)$
                    TSA = TSA + GB(I)*GB(I)$
E16                                  END$
               GBS = TS $
               TSA = SQRT( TSA )$
               WRITE (GBS,TSA)$
               IF ABS( GBS/ ( S(0)*TSA) ) LSS 5.0&-3 THEN BEGIN
B17                 WRITE ('LINEAR MINIMUM IS FOUND IN PART 1 OF LINEAR SEARCH')$
                    WRITE (GBS/( S(0)*TSA)    )$
                    T = H $
FR     E17          GO TO LSA$   END$
                    IF B7 AND   ABS( (TA - H)/H) LSS 5.0&-7 THEN BEGIN
B18                      WRITE ('TIGHT FIT INDEED',TA - H )$
FR     E18               GO TO LSA $ END$
               IF GBS GTR 0.0 THEN BEGIN
B19                 IF B6 THEN BEGIN
B20                      A = 0.0 $
                         FOR I = (1,1,N) DO GA(I) = G(I)$
                         FA = F1 $
FR     E20               GO TO RELOC $    END   ELSE
                    BEGIN
B21                      A = TA $
                         FA = TFA $
                         FOR I = (1,1,N) DO GA(I) = TGA(I)$
                         GAS = TGAS $
FR     E21               GO TO RELOC $    END$
E19                           END$
               T = H $
               TA = H $
               TFA = FB $
               H = MAX( 1.0010*GBS*(T2 - H)/(GBS - G2) + H , H + T)$
               B6 = FALSE $
               T2 = TA $
               G2 = GBS $
               FOR I = (1,1,N) DO TGA(I) = GB(I)$
               TGAS = GBS $
               GO TO INCREASE$



               RELOC:
               IF B4 THEN BEGIN
B22                 WRITE ('USING SLOPE INTERCEPT REFIT')$
                    B4 = FALSE $
                    T = ABS(GAS)*(B - A)/(  ABS(GAS) + ABS(GBS)) + A $
FR                  GO TO L4B $
E22                           END$
               Z = 3.0*(FA -FB)/(B-A) + GAS + GBS $
               W = SQRT( Z*Z - GAS*GBS )$
```

```
         T = B - (((GBS + W - Z)/(GBS - GAS + 2.0*W))*(B-A))$



         L4B:
         FOR I = (1,1,N) DO XEM(I) = X(I) - T*S(I)$
         FP( ROW,XEM,GXEM,FEM,B1,B2,B3,CHEK )$
         IF ROW EQL 3 THEN BEGIN
B23           FOR I = (1,1,N) DO BEGIN
B24                X(I) = XEM(I)$
E24                G(I) = GXEM(I)$    END$
                   WRITE ('RESIDUALS ARE SATISFIED IN PART 2 OF LINEAR SEARCH')$
                   F1 = FEM $
                   CTRD = 0 $
                   CTRB = CTRC $
                   B5 = FALSE $
                   GO TO UPTOP $
E23                    END$



         GS = AGXEM = AG = 0.0 $
         FOR I = (1,1,N) DO BEGIN
B25           GS = GS - GXEM(I)*S(I)$
              AGXEM = AGXEM + GXEM(I)*GXEM(I)$
E25                  END$
         TEST = GS /(    S(0)*SQRT(AGXEM ) )$
         WRITE (FORA,T,FEM,TEST,GS)$
         IF FEM GTR MAX ( FA,FB) OR ABS(TEST) GTR 1.0@-2 THEN BEGIN
B26           RELOCS = RELOCS + 1 $
FR            IF RELOCS EQL 6 THEN GO TO L5 $
              TS = MIN(FA,FB,F1)$
              IF ABS( (FEM - TS)/FEM) LSS 5.0@-7 THEN B4 = TRUE $
              IF GS LSS 0.0 THEN BEGIN
B27                FA = FEM $
                   A = T $
                   GAS = GS $
                   FOR I = (1,1,N) DO GA(I) = GXEM(I)$
E27                GO TO RELOC $ END ELSE
              BEGIN
B28                FB = FEM $
                   B = T $
                   GBS = GS $
                   FOR I = (1,1,N) DO GB(I)=GXEM(I)$
E28                GO TO RELOC $         END$
E26           END$
FR       GO TO L8 $



         L5A:
         FOR I = (1,1,N) DO BEGIN
B29           XEM(I) = XX(I)$
              GXEM(I) = GB(I)$
```

```
E29                      ENDS
      AGXEM = TSA $
      FEM = FB $



      L5:
      WRITE ('XEM')$ WRITE (FORA, XEM)$
      WRITE (' ','FEM')$ WRITE (FORA,FEM)$
      WRITE (' ','GXEM')$ WRITE (FORA,GXEM)$
      WRITE (' ',QQ,CTRA)$
      QG = CTRA*CTRA*COST/( TM*(CTRA + 1)**2)$
      TM = TD = 0.0 $
      FOR I = (1,1,N) DO
            IF ABS( XEM(I) ) GTR TM THEN
                  TM = ABS( XEM(I) )$
      TZ = (F1 - FEM)*3.087 $
      TS = SQRT ( AGXEM )$
      FOR I = (1,1,N) DO
            IF ABS( XEM(I) - X(I) )*G(I)/TM GTR TD THEN
                  TD = ABS( XEM(I) - X(I))*G(I) / TM $
      WRITE (FORA , TS , TD , TZ , F1 )$
      IF TS LSS 1.0S-6 OR
            TD LSS 3.0S-8 OR
                  TZ LSS F1 THEN BEGIN
B30   FR    E30            GO TO L3 $   ENDS
FR    IF BS THEN GO TO L6 $



      COMMENT AVERAGE OF THE LAST THREE VALUES OF TD COMPARED TO FRACTION
            OF MAXIMUM TD ENCOUNTERED $
      IF D2 EQL 0.0 THEN D1 = D2 = TD $
      D3 = D2 $
      D2 = D1 $
      D1 = TD $
      EPA = MAX( TD*1.0S-4,EPA)$
FR    IF  (D1 + D2 + D3)/3.0 LEQ EPA THEN GO TO L3 $



      COMMENT AVERAGE OF LAST THREE PERCENT CHANGE OF THE FUNCTION VALUE F
            COMPARED TO MAXIMUM PERCENT CHANGE. LIMITED BY 0.0001 AND 0.000025
      EX = (F1 - FEM)/FEM $


      IF EX LSS 0.00075 THEN CTRE = CTRE + 1 ELSE CTRE = 0 $
FR    IF CTRE GEQ 8 THEN GO TO L3$

      IF M2 EQL 0.0 THEN M1 = M2 = EX $
      M3 = M2 $
      M2 = M1 $
      M1 = EX    $
      EG = MAX( EG,EX*1.0S-3)$
```

```
                EPB = MIN(EPB, FG)$
                EPB = MAX( EPB,2.0&-5)$
FR              IF M1 + M2 + M3 LSS 3.0*EPB THEN GO TO L3 $



                L0:
                WRITE ('TIME FOR CYCLE'' CLOK -CTIME)           $
                WRITE ('TIME ELAPSED''CLOK - TIME)$
                CTIME = CLOK $
                SGEN1 ( N, S, G , GXEM)$
                TS = 0.0 $
                FOR I = (1,1,N) DO BEGIN
B31                   TS = TS + S(I)*S(I)$
                      G(I) = GXEM(I)$
                      X(I) = XEM(I)$
E31                          END$
                S(0) = SQRT(TS)$
                F1 = FEM$
                GO TO L2$



                L3:
                WRITE ('MINIMUM'' XEM,FEM)$
                WRITE ('TIME'' CLOK - TIME)$
                P5 = FALSE $
                CHEK = & $
                WRITE (FORA,EPA,EPB,E3)$
                FOR I = (1,1,N) DO X(I) = XEM(I)$
                D1 = D2 = D3 = M1 = M2 = M3 = 0.0 $
                EPB = 1.0&-4$
                EG = EPA = 0.0 $
                QG = 1.0 $
                CTRC = 0 $
                CTRE = CTRC $
                FP( ROW,X,G,F1,B1,E2,B3,CHEK )$
FR              IF G(0) EQL 101.0 THEN GO TO COMPLETT$
                IF ROW EQL 0 THEN BEGIN
B32                   WRITE (FORA,F1)$
                      ROW = 0.0 $
                      GO TO UPTOP$
E32                                 END$
                CHEK = &3$
                FP( ROW,X,G,F1,B1,B2,B3, CHEK)$
                WRITE (FORA , F1)$
                GO TO UPTOP $



                COMPLETT:
E1              END$
END BLOCK 2   LEVEL 2
```